ESABASE2 - Framework

Software User Manual

Contract No:	16852/02/NL/JA
Title:	PC Version of DEBRIS Impact Analysis Tool
ESA Technical Officer:	G. Drolshagen, J. Sørensen
Prime Contractor:	etamax space GmbH
Authors:	K. Ruhl, K.D. Bunte, A. Gäde, P. Hake, M. Zaake, J. Weiland, J. Pervez, A. Miller
Date:	2013-07-23
Reference:	R077-230rep_01_08_02_Software_User_Manual_Framework.doc
Revision:	1.8.2
Status:	Final
Confidentiality:	Public

etamax space GmbH
Frankfurter Str. 3d
D-38122 Braunschweig
Germany

Tel.: +49 (0)531.866688.33
Fax: +49 (0)531.866688.99
email: esabase2@etamax.de
http://www.etamax.de

© etamax space GmbH

Table of Contents

Document Information **6**

I. Release Note 6

II. Revision History 6

III. Distribution List 7

IV. List of References 8

V. Glossary 9

VI. List of Abbreviations 9

VII. List of Figures 10

VIII. List of Tables 14

1 Introduction **16**

1.1 The Big Picture 16

1.2 Chapter Overview 17

1.3 Problem Reporting 17

2 Getting Started with ESABASE2 ..... **18**

2.1 Installation and Start 18

2.1.1 Hardware Requirements 18

2.1.2 Installation 19

2.1.3 Operating System Adaptations 20

2.1.4 Upgrade 21

2.1.5 Starting ESABASE2 22

2.2 Basic GUI Usage 23

2.2.1 GUI Overview 23

2.2.2 GUI Adaptation 26

2.2.3 Project Organisation 27

3 Mission and Spacecraft Definition **31**

3.1 Mission 31

3.1.1 Mission File 32

3.1.2 Mission Editor 33
3.1.3 Mission Outline and Properties View 36
3.1.4 Mission Visualisation 37
3.1.5 Ground Track 38
3.1.6 Import of INP Files 39
3.1.7 Trajectory File 40
3.1.8 Orbit Propagator 42
3.2 Geometry 46
3.2.1 Geometry File 47
3.2.2 Geometry Viewing 48
3.2.3 Geometry Editing 53
3.2.4 Geometry Outline and Properties View 76
3.2.5 Boolean Operations 84
3.2.6 Geometry Model Import Interfaces 86
3.2.7 Geometry Model Export Interfaces 93
3.3 Kinematics 99
3.3.1 Bodies 100
3.3.2 Degrees of Freedom 100
3.3.3 The Central Body 101
3.3.4 Articulated Bodies 101
3.4 Pointing 102
3.4.1 Pointing Introduction 102
3.4.2 Pointing Parameters 104
3.4.3 Relationship with Mission Specification 110
3.4.4 Pointing the Bodies and Indeterminacies 113
3.4.5 Pointing Output 116
3.4.6 Pointing Calculation Method 117
4 Solvers 119
5 Troubleshooting 120
5.1 Precautions 120

5.2 Trouble Information 120
5.3 Program Messages 121
5.3.1 Messages: ESABASE2 121
5.3.2 Messages: Eclipse and JVM 122
5.3.3 Messages: Solver 122
5.4 Known Issues 124
5.4.1 Import of Complex BAS Files 124
5.4.2 Memory Leak caused by Open CASCADE 125
5.4.3 Meshing 125
5.4.4 ATI Graphics Card 127
6 Annex A: Geometry Shapes 128
6.1 Basic 3D Shapes 128
6.1.1 Box 129
6.1.2 Cone 132
6.1.3 Cylinder 135
6.1.4 Ellipsoid 138
6.1.5 Paraboloid 142
6.1.6 Sphere 146
6.2 Basic 2D Shapes 150
6.2.1 Disc 151
6.2.2 Plate 153
6.3 Specific Shapes 155
6.3.1 Facet 156
6.3.2 Prism 157
6.3.3 Surface 159
6.4 Complex Shapes 161
6.4.1 Beam 162
6.4.2 Pipe 164
6.4.3 Tank 166
6.4.4 Truss 169

7 Annex B: Additional GUI Elements 174

7.1 Application Menubar 174

7.2 Geometry Editor Toolbar 176

Document Information

I. Release Note

	Name	Function	Date	Signature
Established by:	K. Ruhl, K.D. Bunte, A. Gäde, P. Hake, M. Zaake, J. Weiland, J. Pervez, A. Miller, M. Düttmann	Project Engineers	2013-07-23	
Released by:	K.D. Bunte	Project Manager	2013-04-23	

II. Revision History

Version	Date	Initials	Changed	Reason for Revision
0.1	2009-06-12	KR	All	Taken over from ESABASE2/Debris SUM, restructured chapters, and rewrote introduction and getting started chapters.
0.2	2009-06-22	KR	Chapter 3	Rewrote mission and S/C handling.
0.3	2009-07-06	KR	Chapters 4, 5; Annex A and B	Rewrote debris/meteoroids chapter, troubleshooting section, Annex A and B
0.4	2009-07-28	KR	Chapter 2, 3, 4	Enhancements after PM5.
0.5	2009-08-03	KR	All	Split into Framework, Solver documents
0.9	2009-09-23	KB	All	Review for Final Draft
1.0	2009-09-28	KR	All	Adapted to review results
1.1	2009-10-22	KR, KB	Chapters 2, A	User feedback recognition.
1.2	2009-12-11	AG, KR	Chapters 3, 5	Enclosures (3.2.4), BAS import (5.4.2), Surface nodes (3.2.3.2), orbital points (3.1.2).
1.3	2011-02-03	PH	All	Update of the SUM to reflect the changes introduced with release 4 of ESABASE2.
1.4	2011-03-09	JW	3.2.3.4	Update of tutorial dealing with the Material Editor
1.4	2011-03-30	MZ	3.2.7	GDML export section added
1.5	2011-09-08	MD	3.1.5	Introduced chapter for Ground Track Page.
1.6	2012-09-20	JP	3.2.7, 5.4.3	Meshing info added.
1.7	2013-04-04	PH, AM	1.1, 3.1, 3.4	Added descriptions for Lunar Missions.
1.8	2013-04-18	AM	3.1.8, 3.4.3	Extended the descriptions for lunar missions.
1.8.1	2013-06-11	AM	3.1.8.3 3.2.6.3 5.4.3	Include the definitions of SRP and drag (SAPRE), Note for GDML import (internet connection), Extended Meshing section
1.9	2013-07-23	MZ	3.2.3.2	Extended Meshing section

III. Distribution List

Institution	Name	Remarks
ESTEC	Gerhard Drolshagen	

IV. List of References

/1/ ESABASE2 homepage, http://www.esabase2.net/

/2/ K. Ruhl, K.D. Bunte, ESABASE2/Debris software user manual, R077-232rep, ESA/ESTEC Contract 16852/02/NL/JA "PC Version of DEBRIS Impact Analysis Tool", etamax space, 2009

/3/ K. Ruhl, A. Gaede, ESABASE2/Atmosphere-Ionosphere software user manual, R077-233rep, ESA/ESTEC Contract 21739/08/NL/JK "Rapid Prototyping Toolkit for Space Environment Engineering Tools", etamax space, 2009

/4/ K. Ruhl, J. Weiland, ESABASE2/COMOVA software user manual, R077-234rep, ESA/ESTEC Contract 21739/08/NL/JK "Rapid Prototyping Toolkit for Space Environment Engineering Tools", etamax space, 2009

/5/ K. Ruhl, A. Gaede, ESABASE2/Sunlight software user manual, R077-235rep, ESA/ESTEC Contract 21739/08/NL/JK "Rapid Prototyping Toolkit for Space Environment Engineering Tools", etamax space, 2009

/6/ ESABASE User Manual, ESABASE/GEN-UM-070, Issue 1, Mathematics & Software Division, ESTEC, March 1994

/7/ SWENET, ESA's Space Weather European Network, since 2004, http://www.esa-spaceweather.net/swenet/

/8/ STEP file format, http://en.wikipedia.org/wiki/ISO_10303

/9/ Netgen mesher user manual, http://www.mathcces.rwth-aachen.de/netgen/doku.php/docu/ngusermanual

/10/ P. Hake, M. Zaake, Data Exchange Methods for Space Environment Effect Tools design definition file, R089_221rep, ESA/ESTEC contract 4000101301 "Data Exchange Methods for Space Environment Effect Tools", etamax space, 2010

/11/ SPENVIS: The Space Environment Information System, orbit generator description, http://www.spenvis.oma.be/help/models/sapre_earth.html

V. Glossary

Term	Description
Application	ESABASE application such as e.g. the "Debris" or the "Sunlight" application.
Eclipse	Eclipse is an open source community whose projects are focused on providing an extensible development platform and application frameworks for building software. For detailed information refer to http://www.eclipse.org/.
ESABASE	Unix-based analysis software for various space applications. For details refer to the ESABASE User Manual /6/.
ESABASE2	New ESABASE version running on PC-based Windows platforms (to be distinguished from the "old" Unix-based ESABASE).
Geometric(al) (analysis)	Analysis of a full geometric model.
Georelay	Object pointing keyword: tracking of a GEO satellite.
Non-geometric(al) (analysis)	Analysis of a plate; faster than analysing a full geometrical model.
STEP	Acronym which stands for the Standard for the Exchange of Product model data /8/.

VI. List of Abbreviations

Abbreviation	Description
GUI	Graphical User Interface
JVM	Java Virtual Machine
NASA	National Astronautics and Space Administration
OCAF	Open CASCADE Application Framework (contains the ESABASE2 data model)
RTP	Randomly Tumbling Plate

VII. List of Figures

Figure 2-1: ESABASE2 directory tree 19
Figure 2-2: ESABASE2 directory tree: install supplements 20
Figure 2-3: ESABASE2 user interface 22
Figure 2-4: User interface overview 23
Figure 2-5: Menubar and Toolbar 24
Figure 2-6: Explorer, Outline, Properties view 25
Figure 2-7: Common View buttons 26
Figure 2-8: "New Project" wizard, page 1 27
Figure 2-9: "New Project" wizard, page 2 28
Figure 2-10: A new project and its three default files 28
Figure 2-11: Creating new files within a project 29
Figure 2-12: Creating folders within a project, using the context menu 30
Figure 3-1: Mission file creation wizard 32
Figure 3-2: Mission Editor, Outline and Properties view 33
Figure 3-3: Mission Editor, with Pertubations in Outline. 36
Figure 3-4: Mission Editor, Visualisation page 37
Figure 3-5: Mission Editor, Ground track page 38
Figure 3-6: Mission file creation wizard, for INP files 39
Figure 3-7: Trajectory File 40
Figure 3-8: Geometry file creation wizard 47
Figure 3-9: Geometry editor, empty file 48
Figure 3-10: Geometry editor, first box 49
Figure 3-11: Geometry button, Toolbar (right side), Mesh button 49
Figure 3-12: Geometry Editor in meshed view 50
Figure 3-13: Geometry editor, Toolbar (left side) 51
Figure 3-14: Geometry editor, example satellite 53
Figure 3-15: Geometry editor, Box wizard, Size page 55
Figure 3-16: Geometry editor, Box wizard, Meshing page 56

Figure 3-17: Geometry editor, Box wizard, Meshing page, Raytracing and sides 57
Figure 3-18: Geometry editor, transparency set to 5 58
Figure 3-19: Geometry editor, default Cone 60
Figure 3-20: Geometry editor, Cone wizard, Position and Attitude page 61
Figure 3-21: Geometry editor, Wizard header 62
Figure 3-22: Geometry editor, Material page 63
Figure 3-23: Material Overview Page: Edit materials button 64
Figure 3-24: Material Editing Page 64
Figure 3-25: Material editing page: Properties view 65
Figure 3-26: Material editing window, Filter button 67
Figure 3-27: Material overview page, select a material for all surfaces 68
Figure 3-28: Geometry editor, example satellite with antenna dish at bottom 69
Figure 3-29: Geometry editor, Pointing page (antenna dish) 70
Figure 3-30: Geometry editor, Kinematic page (antenna dish) 71
Figure 3-31: Geometry editor, Pointing page (System node) 72
Figure 3-32: Geometry editor, Kinematic page (solar panel) 73
Figure 3-33: Geometry editor, Pointing page (solar panel) 74
Figure 3-34: Geometry editor, example satellite with two solar panels 75
Figure 3-35: Orbital point configurations, example satellite 75
Figure 3-36: Geometry editor, Outline, Cone selected 76
Figure 3-37: Geometry editor, Outline, Cone renaming 77
Figure 3-38: Geometry editor, Outline, node renaming 78
Figure 3-39: Geometry editor, Outline, Copy & Paste 79
Figure 3-40: Geometry editor, Outline, expanded 80
Figure 3-41: Geometry editor, Outline and Properties view 81
Figure 3-42: Geometry editor, Outline and Properties view, editing 82
Figure 3-43: Geometry editor, Outline and Properties view, Enclosure definition 83
Figure 3-44: Geometry editor, Outline and Properties view, Enclosure definition 84
Figure 3-45: The context menu in the outline, containing the BO 85
Figure 3-46: The shape after performing the operation 85

Figure 3-47: Modifying a BO via the outline 85
Figure 3-48: The shape after the modification 86
Figure 3-49: Geometry file creation wizard, STEP variant 87
Figure 3-50: Geometry editor, imported STEP ATV model 88
Figure 3-51: Geometry file creation wizard, BAS variant 89
Figure 3-52: Geometry editor, imported HUBBLE.BAS 90
Figure 3-53: Geometry file creation wizard, GDML variant 91
Figure 3-54: Geometry file creation wizard, STEP-SPE variant 92
Figure 3-55: The ATV imported from a BAS file 93
Figure 3-56: The GDML export entry in the context menu 94
Figure 3-57: The directory for the export is selected 94
Figure 3-58: Files generated during GDML export 95
Figure 3-59: Typical content of "mainFile.gdml" 95
Figure 3-60: A simple satellite structure to demonstrate the STEP-SPE export 96
Figure 3-61: STEP-SPE export in the outline 97
Figure 3-62: STEP export in the outline 98
Figure 3-63: Spacecraft model 103
Figure 3-64: Pointing realisation 113
Figure 3-65: Attitude indeterminacy (central body) 114
Figure 3-66: Attitude indeterminacy (2 degrees of freedom) 115
Figure 5-1: Troubleshooting, Messages, ESABASE2 Debris analysis 121
Figure 5-2: Troubleshooting, Messages, Eclipse and JVM abnormal termination 122
Figure 5-3: Troubleshooting, Messages, Solver warning (kinematic) 123
Figure 5-4: Application of the ESABASE mesher in the "Outline" 126
Figure 6-1: Basic 3D geometry shapes 128
Figure 6-2: Parameters of a box object in ESABASE (left) and GDML (right) style 129
Figure 6-3: Outer shape parameters of a cone object in ESABASE (left) and GDML (right) style 132
Figure 6-4: Parameters of a cylinder object in ESABASE (left) and GDML (right) style 135
Figure 6-5: Meshing parameters of the cylinder 137

Figure 6-6: Parameters of an ellipsoid object in ESABASE (left) and GMDL (right) style 138
Figure 6-7: Meshing parameters of the ellipsoid 141
Figure 6-8: Parameters of a paraboloid object 142
Figure 6-9: Meshing parameters and element numbering of the paraboloid 145
Figure 6-10: Parameters of a sphere object in ESABASE (left) and GDML (right) style 146
Figure 6-11: Meshing parameters of the sphere 149
Figure 6-12: Basic 2D geometry shapes 150
Figure 6-13: Parameters of a disc object 151
Figure 6-14: Parameters of a plate object 153
Figure 6-15: Meshing parameters of the PLATE 154
Figure 6-16: Specific geometry shapes 155
Figure 6-17: Parameters of a prism object 157
Figure 6-18: Parameters of a surface object 159
Figure 6-19: Complex geometry shapes 161
Figure 6-20: Parameters of a beam object 162
Figure 6-21: Parameters of a pipe object 164
Figure 6-22: Parameters of a tank object 166
Figure 6-23: Meshing parameters of the tank 167
Figure 6-24: Overview of the truss object 169
Figure 6-25: Points of the truss object 169
Figure 6-26: Additional parameters of the truss object 170
Figure 6-27: Overview of the four sided truss object 171
Figure 6-28: Points of the four sided truss object 172
Figure 6-29: Additional parameters of the four sided truss object 172
Figure 7-1: ESABASE2 menu bar 174
Figure 7-2: ESABASE2 help system 175
Figure 7-3: Geometry editor, Toolbar (right side) 176

VIII. List of Tables

Table 2.1: Hardware Requirements 18

Table 3.1: Orbit input parameters 44

Table 3.2: Examples for Degree of Freedom specifications 100

Table 3.3: The "Inherit parent values" parameter 106

Table 3.4: The "VEC1" parameter 107

Table 3.5: The "VEC2" parameter 107

Table 3.6: The "pdir" parameter 108

Table 3.7: The "track" parameter 109

Table 3.8: The "inverse" parameter 109

Table 5.9: Overview of the capabilities of the ESABASE2 meshers 127

Table 6.1: Definition of the BOX shape in ESABASE style 129

Table 6.2: Definition of the BOX shape in GDML style 130

Table 6.3: Definition of the CONE shape in ESABASE style 133

Table 6.4: Definition of the CONE shape in GDML style 133

Table 6.5: Definition of the CYLINDER shape in ESABASE style 136

Table 6.6: Definition of the CYLINDER shape in GDML style 136

Table 6.7: Definition of the ELLIPSOID shape in ESABASE style 139

Table 6.8: Definition of the ELLIPSOID shape GDML style 139

Table 6.9: Definition of the PARABOLOID shape 143

Table 6.10: Definition of the SPHERE shape in ESABASE style 147

Table 6.11: Definition of the SPHERE shape in GDML style 147

Table 6.12: Definition of the DISC shape 152

Table 6.13: Definition of the PLATE shape 153

Table 6.14: Definition of the FACET dummy shape 156

Table 6.15: Definition of the PRISM shape 157

Table 6.16: Definition of the SURFACE shape 159

Table 6.17: Definition of BEAM shape 162

Table 6.18: Definition of the PIPE shape 164

Table 6.19: Definition of the TANK shape 166
Table 6.20: Definition of the TRUSS3 shape.....170
Table 6.21: Definition of the TRUSS4 shape.....173

1 Introduction

Before you dive into the ESABASE2 analysis work, this short introduction will provide you with a birds-eye-view understanding, with the help of the following items:

- *The Big Picture*: Where does ESABASE2 stand?
- *Chapter overview*: So you find your way through the chapters more quickly.
- *Problem Reporting*: What to do if you encounter troubles.

1.1 The Big Picture

ESABASE2 is a software application (and framework) for space environment analyses, which play a vital role in spacecraft mission planning. Currently (2013), it encompasses Debris/meteoroid /2/, Atmosphere/ionosphere /3/, Contamination/outgassing /4/ and Sunlight /5/ analyses; with this, it complements other aspects of mission planning like thermal or power generator design.

In CCN 9, the Debris application was extended to allow the analysis of lunar missions.

The application grew from ESABASE2/Debris, an application for space debris and micrometeoroid impact and damage analysis, which in turn is based on the original ESABASE/Debris software /6/ developed by different companies under ESA contract. ESABASE2 adds a modern graphical user interface enabling the user to interactively establish and manipulate three-dimensional spacecraft models and to display the selected orbit. Analysis results can be displayed by means of the colour-coded surfaces of the 3D spacecraft model, and by means of various diagrams.

The development of ESABASE2 was undertaken by etamax space GmbH under the European Space Agency contract No. 16852/02/NL/JA. The first goal was to port ESABASE/Debris and its framework/user interface to the PC platform (Microsoft Windows) and to create a modern user interface.

From the start, the software architecture has been expressively designed to accommodate further applications: the solvers outlined in the first paragraph were added, and more modules like e.g. Radiation are to follow.

ESABASE2 is written in Fortran 77, ANSI C++ and Java 6. The GUI is built on top of the Eclipse rich client platform, with 3D visualisation and STEP import realised by Open CASCADE. Report and graphs are based on the JFreeReport/JFreeChart libraries.

1.2 Chapter Overview

This software user manual is intended to be read by persons running the program and interpreting or using the results of ESABASE2. It contains the following chapters:

- Chapter 1, *Introduction*: You are reading it right now.
- Chapter 2, *Getting Started*: How to install, start and use the software, without going into particular solvers (debris, atmosphere, ionosphere).
- Chapter 3, *Mission and Spacecraft Definition*: How to specify the basics: mission parameters and spacecraft geometry. Can be re-used with all solvers.
- Chapter 4, *Solvers*: Pointers to the solver handbooks, e.g. Debris or Sunlight.
- Chapter 5, *Troubleshooting*: What to do in case of errors.

Each solver (e.g. Debris /2/, Atmosphere/Ionosphere /3/, Sunlight /5/) has its own handbook; this manual explains only the functionality common to all.

This user manual refers to **ESABASE2 version 4.0.0**. Other versions of ESABASE2 (and of course the original Unix implementation of ESABASE) are not covered.

1.3 Problem Reporting

ESABASE2 has been extensively tested. However, errors might occur at different layers of the software (GUI, business layer, data handling). Aside from the GUI messages, the log files in the ESABASE2 logs directory may yield further information.

If a problem cannot be solved, you should contact the ESABASE2 team at etamax space to investigate the issue in detail. Please do not hesitate to visit http://www.esabase2.net for contact information. You will also find a problem reporting form in the User's area of the ESABASE2 website.

2 Getting Started with ESABASE2

This chapter describes how you can install and run ESABASE2 in a basic way. If you have already created a spacecraft geometry and a mission and/or made an analysis, you can safely skip this chapter. It is divided into the following subsections:

- *Installation and Start*: How to install ESABASE2 and start it; you should end with the application on your screen.
- *Basic GUI Usage*: Explanation of the ESABASE2 graphical user interface concept; how to perform tasks that are common to all ESABASE2 solvers.

2.1 Installation and Start

In this section, you will learn how to install and start ESABASE2:

- *Hardware Requirements*: Just to make sure your PC is sufficiently powerful. Especially large geometries and many orbital points can be quite taxing.
- *Installation*: How to install the ESABASE2 software on your harddisk.
- *Operating System Adaptations*: For older operating systems (e.g. Windows 2000), some adaptations are necessary in order to run ESABASE2.
- *Upgrade*: Notes for users upgrading from a previous version of ESABASE2.
- *Starting ESABASE2*: The different ways to start the application, and details about where your files are on the harddisk.

2.1.1 Hardware Requirements

Before installing ESABASE2, you should make sure that the system requirements listed in Table 2.1 are met. Otherwise, large geometries or analysis runs with a large number of orbital points could take very long or fail due to insufficient memory.

	Minimum	**Recommended**
Operating System	Windows 2000	Windows XP, Vista
RAM	1 GByte	4 GBytes
Free Disk Space	1 GByte	2 GBytes
CPU (Clock frequency)	2 GHz single core	2 GHz dual core
Graphics Card	Nvidia Geforce 6200	Nvidia Geforce 9800

Table 2.1: Hardware Requirements

2.1.2 Installation

To install ESABASE2, unzip the ESABASE2_4.0.0.zip archive from the installation DVD to a location of your choice. We recommend deactivating anti-virus software as its file scanning activities can considerably slow down the unpacking process.

If you open a Windows Explorer and navigate to the ESABASE2 application directory, the layout shown in the following figure will be presented to you:

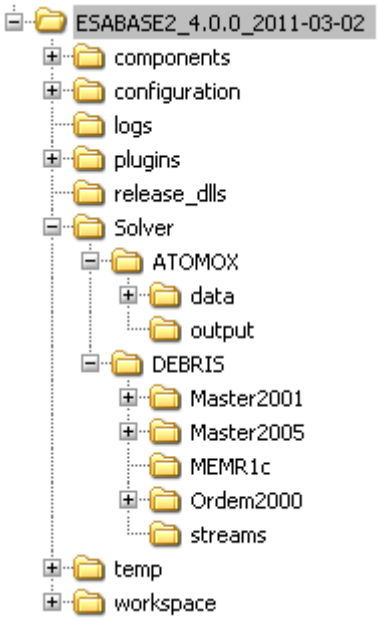

Figure 2-1: ESABASE2 directory tree

In the logs folder, you can find the application log files (named with application start date), useful for troubleshooting. If you report a problem to the etamax team, please attach the pertinent log file.

In case you are an original ESABASE expert, the contents of the Solver directory could be of interest to you. Here, expert users can manipulate population data and data tables. Do so at your own risk, and always keep a backup.

Please do not touch the components, configuration, plugins, and release_dlls folders. Only look into the temp directory for curiosity; do not change anything there.

Once you run the application, a workspace folder will be added. All of your files are saved here.

2.1.3 Operating System Adaptations

The operating system for ESABASE2 is Windows 2000 SP4, XP SP3 or Vista SP1. With the older OS versions, the following components may not be present and must then be installed manually:

- Windows installer (version "KB884016-v2-x86")
- DLLs for generic Windows functions (version "x86_2009_sp2").

To install the missing components, locate the components\install-supplements directory below the ESABASE2 base directory, as shown in the figure below (this is an exception to the "don't touch the components directory" from the previous subsection).

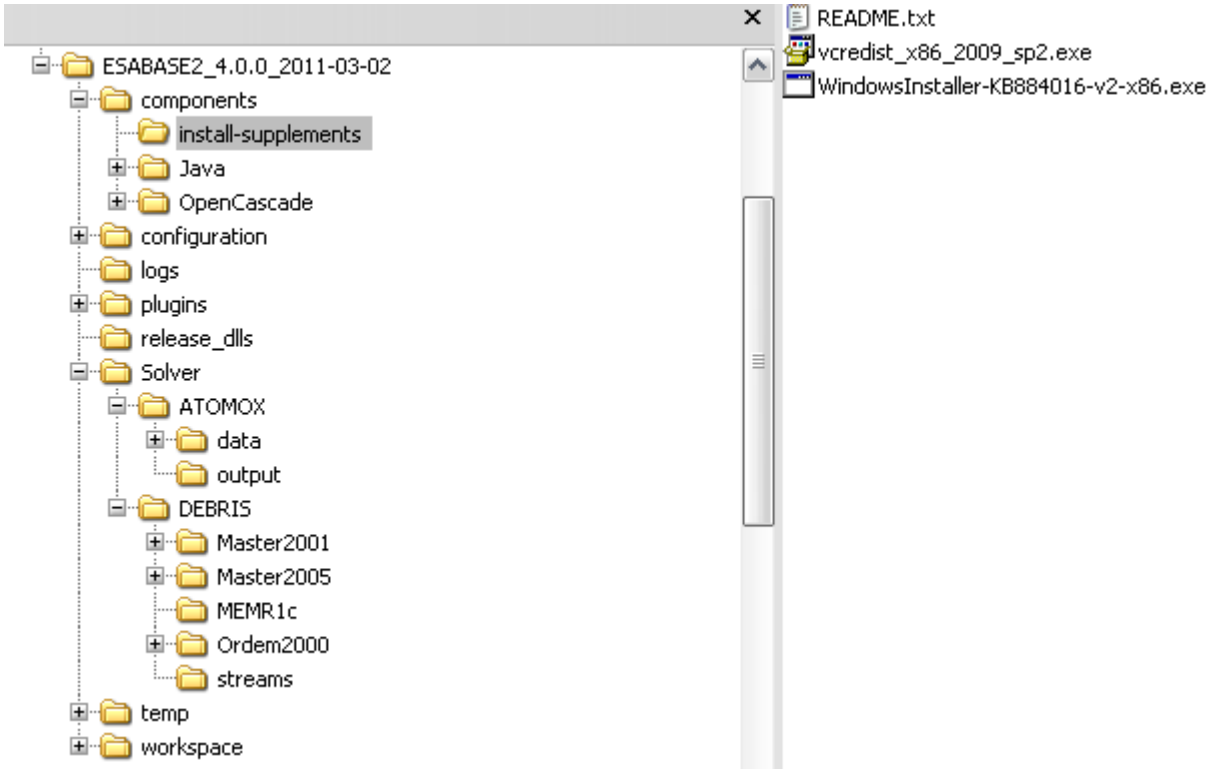

Figure 2-2: ESABASE2 directory tree: install supplements

Doubleclick WindowsInstaller-KB884016-v2-x86.exe and follow the instructions. Then repeat the same procedure for vcredist_x86_2009_sp2.exe. Your system is now equipped with the newest version of central DLLs used by ESABASE2.

2.1.4 Upgrade

If you are a first-time user of ESABASE2, you can safely skip this chapter. If you are upgrading, the following hints may be useful for you.

The new installation is in a separate directory from your old installation. It is possible to have several versions on harddisks; the development team is doing it all the time.

To keep your workspace from a previous version (e.g. ESABASE2 1.4.2), delete the workspace directory from the new application directory, and move your original workspace directory here.

Then start the new application; your workspace should be immediately visible. This is all you need to do to the workspace replacement.

Individual projects can be imported also, by copying the project folder or files to the existing workspace directory. After starting the application the project files should be visible.

Please note that with ESABASE2 version 2.0.0, a new file storing format has been introduced. Files from earlier versions are readable (they are converted automatically to the new version), but it is not possible the other way round.

For example, a geometry file created or modified with ESABASE2 version 4.0.0 will not be readable by ESABASE2 version 1.4.2. It is always possible that a file of an older version can be imported and used in a newer version of an application but not vice versa.

2.1.5 Starting ESABASE2

To launch the ESABASE2 application, double-click the file esabase2.exe in the application directory.

A splash screen will appear to accompany the loading process, after which the main GUI will become visible on screen, as illustrated by the following screenshot.

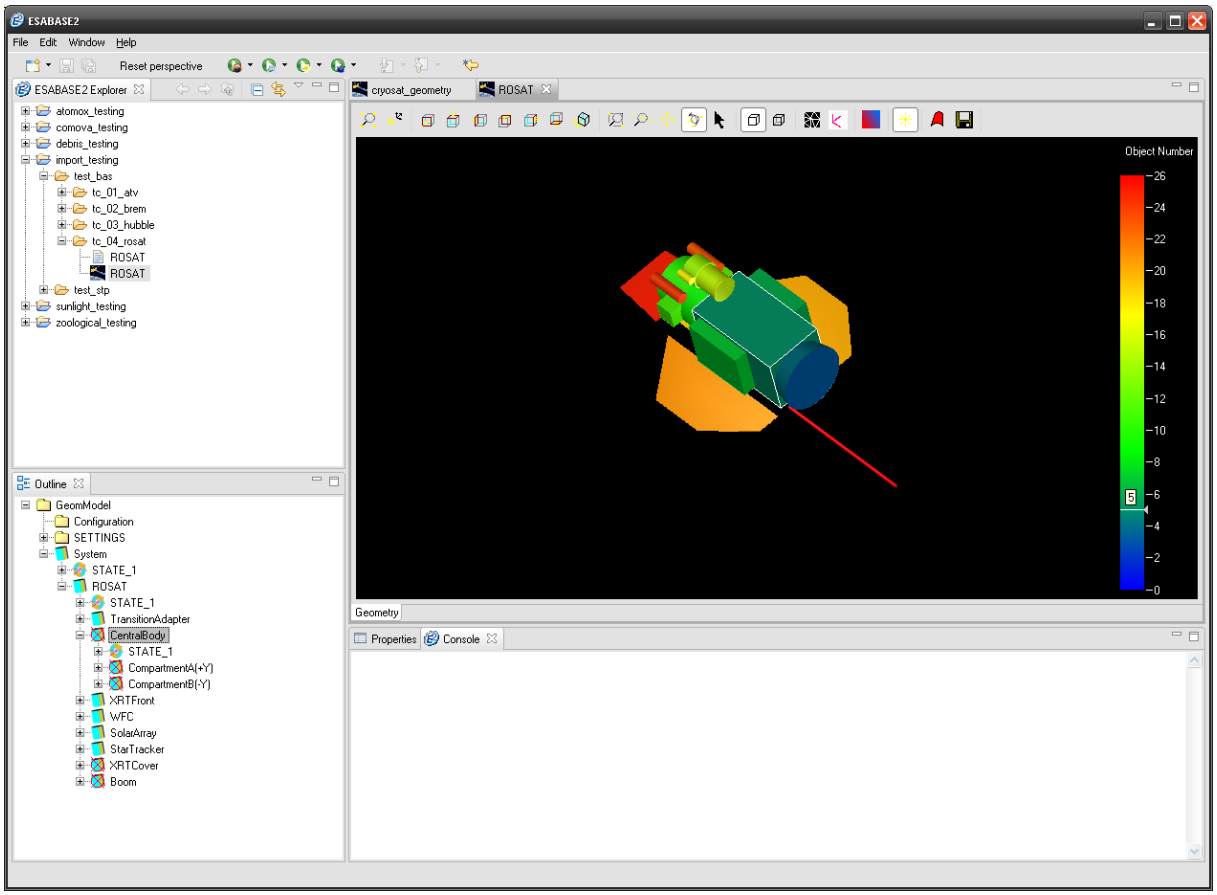

Figure 2-3: ESABASE2 user interface

You will notice that the GUI consists of multiple sub-windows (called "views"). In the figure above, the left column consists of an "ESABASE2 Explorer" and an "Outline"; to the right, most of the place is taken by a spacecraft geometry editor; and at the bottom, you can see a "Properties" view.

The next section will explain the usage of the GUI.

2.2 Basic GUI Usage

After reading this chapter, you will be able to navigate the GUI and make sense of the ESABASE2 workflow, in the following steps:

- *GUI Overview*: Explains the contents of each view within the application, and its purposes.
- *GUI Adaptation*: How to customise the view arrangement to suit your informational needs.
- *Project organisation*: How to create and organise projects and geometry, mission and solver files.

2.2.1 GUI Overview

This subsection will give you an overview of the main GUI parts, enumerated from 1 to 6 in the following figure.

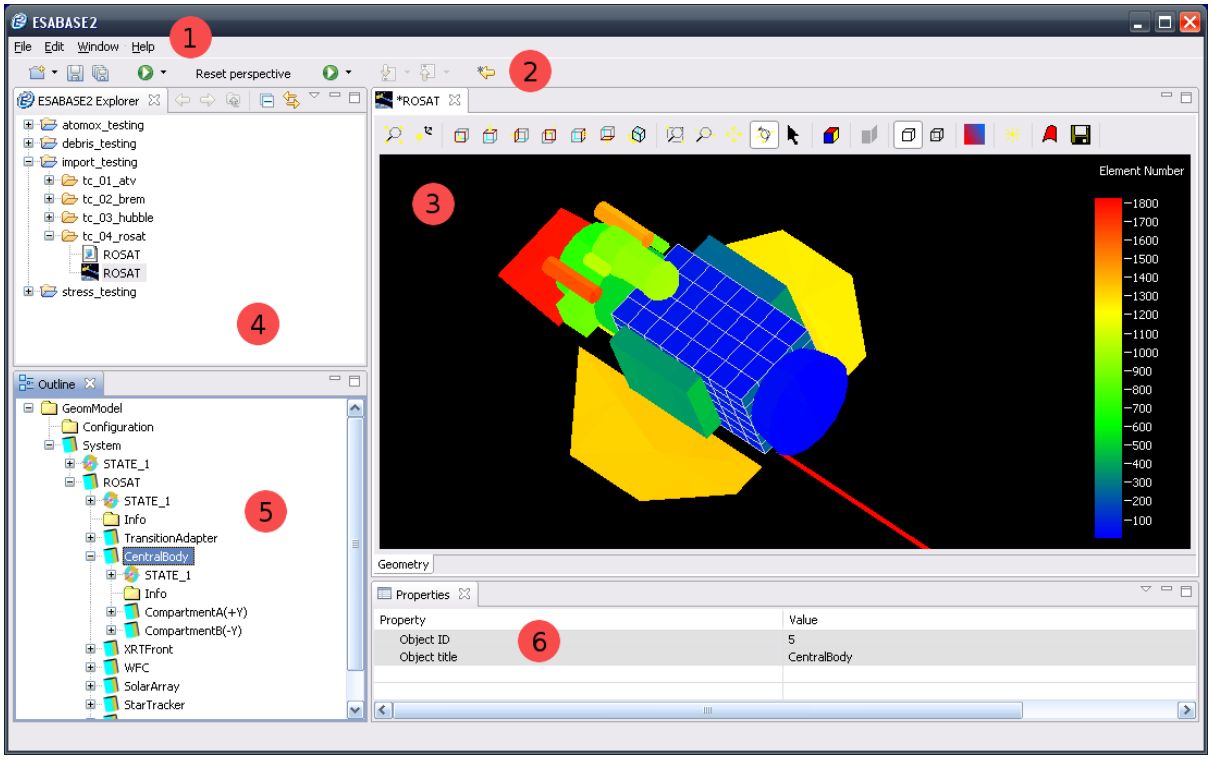

Figure 2-4: User interface overview

There is a menu bar (1), a toolbar (2), an editor (3) and then a number of supporting views (4) to (6).

They serve the following purposes:

- The menu bar gives full (but complex) access to all major operations concerning projects and files.
- The tool bar gives easy (but incomplete) access to the most frequently used functionalities.
- The editor area contains an editor for a file (here, a geometry editor on a ROSAT spacecraft geometry). Each file has its own editor.
- The workspace explorer shows your projects and the folders and the files within it.
- Each file is modelled as a tree structure (think "directories" and "files" like in your file system). The Outline shows the "directory" nodes.
- Complementary, the Properties view shows the "file" nodes.

The menu bar contains "File", "Edit", "Help" entries (probably familiar to you from other applications) and a "Window" menu, which allows you to customize the GUI both through Perspective settings and the Preferences. Please see Annex B for details, and the following figure for overview.

Figure 2-5: Menubar and Toolbar

The toolbar contains "New", "Save / Save All" and "Run" buttons. The "New" (project, geometry, mission, etc.) entry is explained in section 2.2.3 "Project Organisation"; the "Run" entries apply to the different Solvers (e.g. Debris, Sunlight, Atmosphere) and are explained in the respective solver handbooks.

If you hopelessly disarranged the application window, you can press the "Reset Perspective" button as a last resort. It will restore the standard GUI settings.

The "Next / Prev Change" and "Back" buttons are shortcuts to menu entries. They will be removed in future versions of ESABASE2.

The editor area contains the active editors. Whenever there is more than one editor open, tabs will show the list, and the active tab belongs to the active editor. Please see chapter 3 "Mission and Spacecraft" definition, for details.

Explorer, Outline and Properties view give you information about the data contents in a file. This is done in 3 detail levels, as the following figures show.

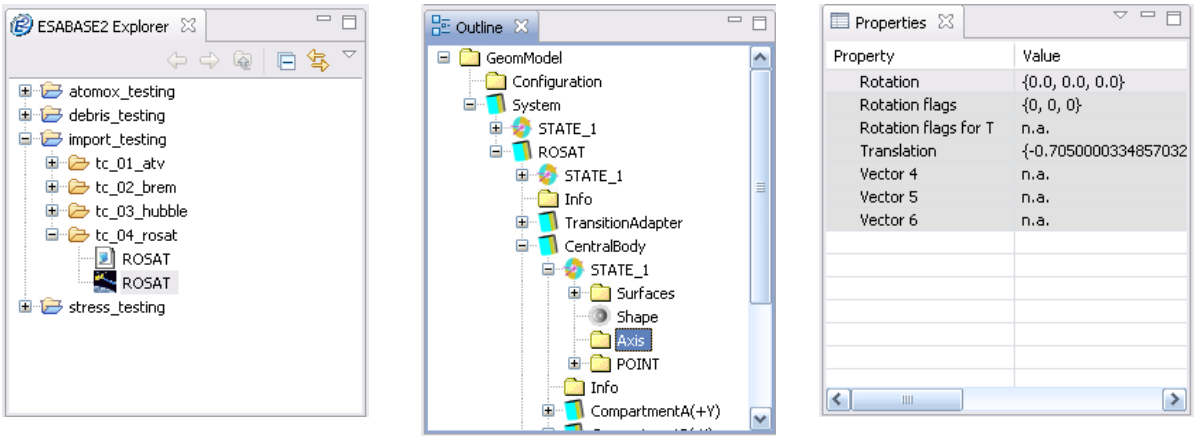

Figure 2-6: Explorer, Outline, Properties view

The Explorer shows the workspace with projects, directories and files; the Outline shows the tree structure of an open file; and the Properties view shows detailed data entries of one specific node.

In the above example, a ROSAT geometry is shown (see Explorer). This in turn consists of a tree with a System and a main body "ROSAT", which has a "CentralBody" with a "STATE_1", describing an "Axis" (see Outline). To see exactly how the axis is specified, refer to the Properties view, which shows e.g. a "Rotation" attribute.

2.2.2 GUI Adaptation

In this section, you will learn how to customise the views of the ESABASE2 application window. If you are already familiar with GUI handling of multi-window applications, you can safely skip this chapter.

The first thing to know is how to manipulate the views within the main application window. For this purpose, each view has a tab with its name, and additional buttons, as shown in the following figure.

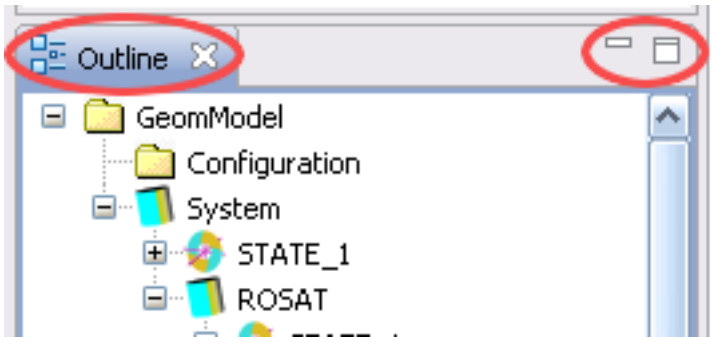

Figure 2-7: Common View buttons

To the right, the first button is to minimise the view, the second one is to maximise it. You can also maximise a view by double clicking its name tab (left). By double clicking the tab of an already maximised view, the size is normalised.

You can also grab the tab and move the view around, and snap it into different positions within the main window. This includes tabbing views, i.e. layering several views on top of each other.

Lastly, the edges of each view can be grabbed for resizing. Depending on the position of the view in the main window, only the view or a whole column is resized.

All of your changes are saved automatically in the current perspective ("arrangement of views"). To reset the perspective to the original settings, choose "Window → Reset Perspective" in the menubar.

2.2.3 Project Organisation

In order to work with ESABASE2, you will work with input and output files. Both are organised in projects. This subsection deals with the following cases:

- Creating a project
- Creating input files (e.g. mission, spacecraft)
- Moving files around

2.2.3.1 Creating a Project

The first thing to do after starting ESABASE2 is to create a new project (which will contain S/C geometry, orbit and solver parameters). We recommend creating one project per mission. To do so, click "File → New → Project" in the menubar; the wizard shown in the following figure will open.

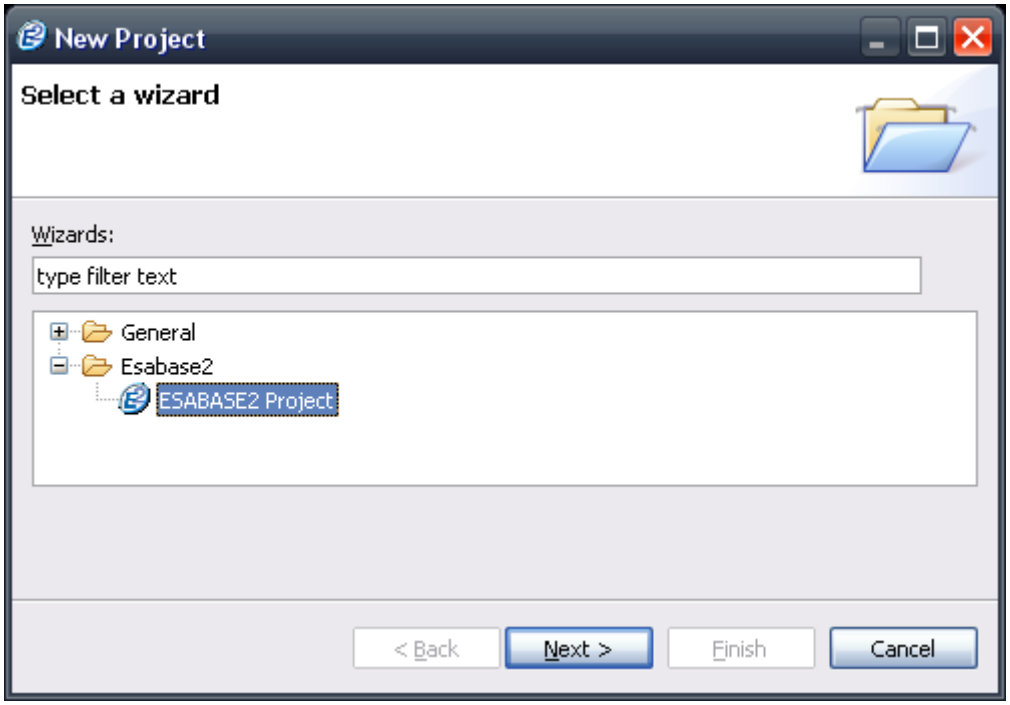

Figure 2-8: "New Project" wizard, page 1

On this first page, you can select the type of project; choose "ESABASE2 Project" (in future versions of ESABASE2, other kinds of project may be possible). Then, click next. This will lead to the second wizard page, illustrated in the following figure:

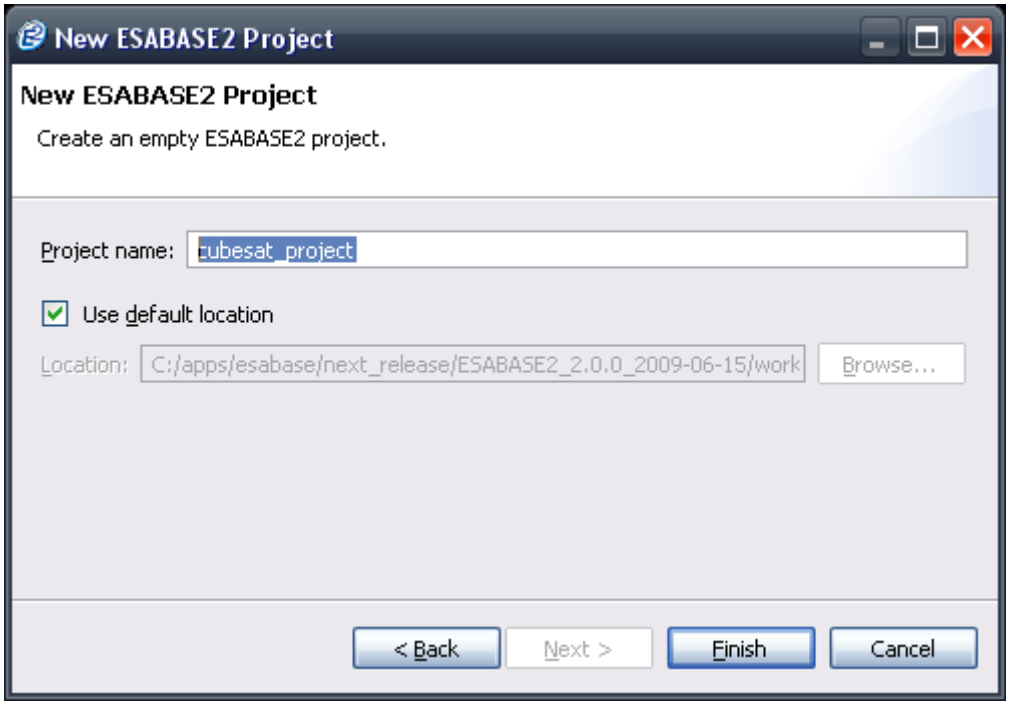

Figure 2-9: "New Project" wizard, page 2

Choose a project name in the first text field; we recommend "<S/C name>_project". Then click the "Finish" button.

When looking at the ESABASE2 Explorer (top-left in the window), you will see that the project has been created with three default files, shown in the following figure:

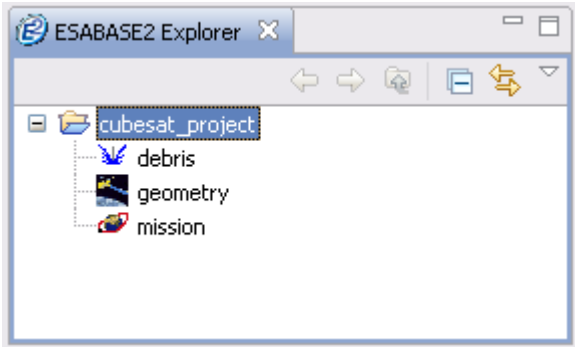

Figure 2-10: A new project and its three default files

The "geometry" file contains the S/C geometry, the "mission" file describes the orbit, and the "debris" solver input file contains parameters for ESABASE2/Debris.

2.2.3.2 Creating Input Files

When you created a project, 3 default files for S/C geometry, mission specification, and Debris solver were created. To create additional files, choose "File → New → <Type>" in the menubar, as illustrated by the following figure:

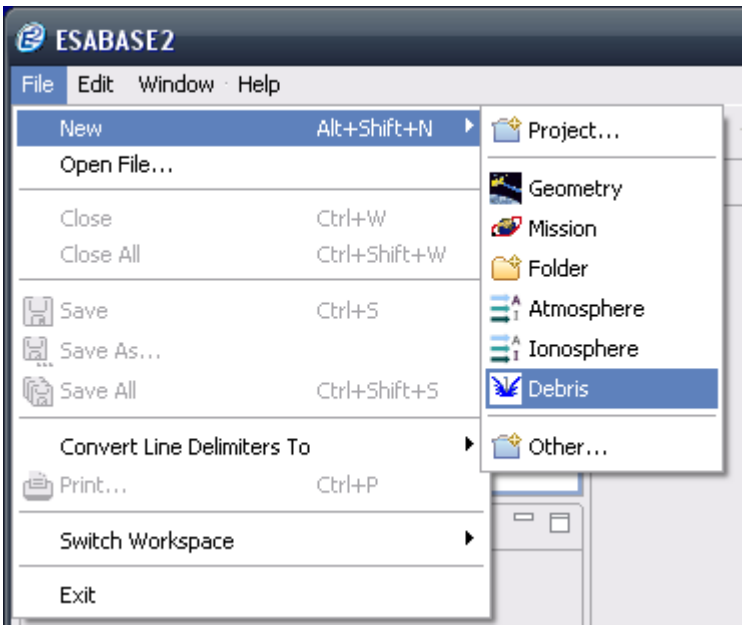

Figure 2-11: Creating new files within a project

All possible input files are listed here. In the example above, you see "New" options for projects (see previous subsection), geometry, mission, folder; and solver input files for Atmosphere/Ionosphere and Debris.

For each file type, an appropriate wizard will be opened; at the minimum, it will ask you for the filename and where to save it in the project structure. Import options, if applicable, will also be presented here; please see the respective chapters for more information.

All newly created input files are filled with default values, such that you can immediately perform a solver run with it. This means that with the 3 default files, you could initiate an ESABASE2/Debris analysis (see Debris solver handbook).

2.2.3.3 Moving Files Around

In larger projects, you may wish to organise your input (and output) files into folders. To do so, rightclick on the project (this will open a context menu, as shown in the following figure) and create a new folder.

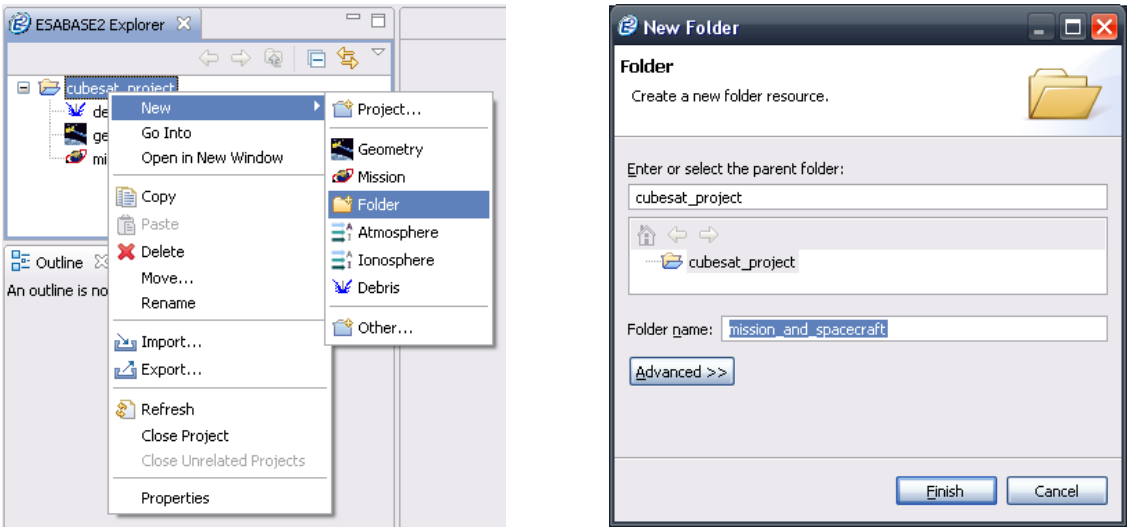

Figure 2-12: Creating folders within a project, using the context menu

A wizard opens, asking you for a directory name (in the example above, the name "mission_and_spacecraft" is chosen). Enter a file name and click "Finish".

To create subfolders, repeat the process on the desired parent folder within the ESABASE2 Explorer (instead of on the project).

To move a file to a different folder, close any editor operating on the file, and then grab (hold left mouse button) the file, move the mouse cursor to the desired folder, and drop (release left mouse button) the file. To move a file to the project root directory, drop the file on the project name.

Drag and drop also works between the ESABASE2 Explorer and the Windows Explorer, e.g. to copy files sent to you by email into the ESABASE2 project.

The specifics of mission and geometry files will be explained in the next section 3; Debris and other solver files will be explained in the solver handbooks.

3 Mission and Spacecraft Definition

Every analysis within ESABASE2 needs a mission specification and a spacecraft geometry definition, as well as a solver-specific input file. In this chapter, you will learn how to specify mission and S/C geometry, in the following steps:

- *Mission*: How to specify the S/C orbit and timeframe of the mission.
- *Geometry*: How to build a 3D spacecraft geometry in a CAD way, or how to import one from BAS or STEP files.
- *Kinematics*: How to supply the S/C geometry with the ability to move its parts, both by rotating and translating subsystems during a mission.
- *Pointing*: How to specify pointing on a spacecraft, both on the system and on sub-components (e.g. solar panels pointing to the sun). Subsystems need kinematics to be able to move.

3.1 Mission

All solvers within ESABASE2 perform their analysis on orbital points, i.e. fixed locations on a given orbit at a given point of time. To determine these orbital points, the mission file is used.

In the following, you will specify a mission in the following steps:

- *Mission File*: How to create a new mission file.
- *Mission Editor*: How to specify the orbit, timeframe, and orbital point parameters.
- *Mission Outline and Properties View*: How to specify additional parameters (for Expert users only).
- *Mission Visualisation*: The mission editor contains a preview page, on which you can see the specified orbit.
- *Import of INP files*: Alternatively, you can import original ESABASE .INP files for the mission specification.
- *Orbit Propagator*: Explains how the orbit propagator works, and its limitations.

We start with the mission editor.

3.1.1 Mission File

With the creation of a project, a new mission file has been created; however you might also want to create a new one. To do so, choose "File → New → Mission" in the menubar. A wizard as shown in the following figure will be opened.

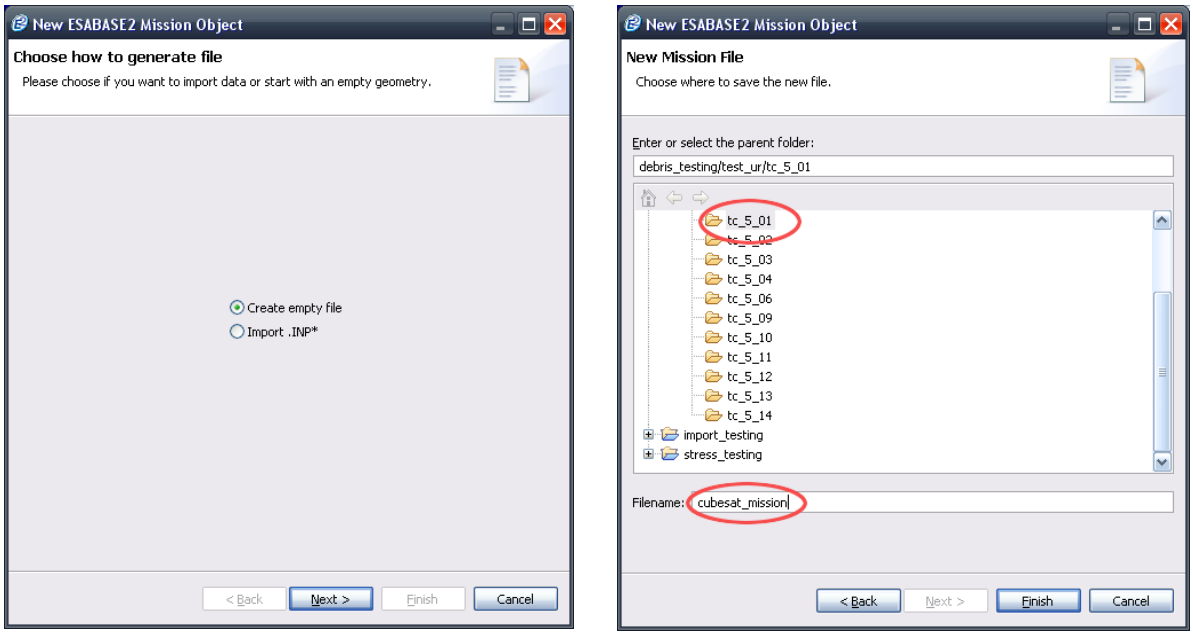

Figure 3-1: Mission file creation wizard

On the first wizard page, "Create empty file" is selected as default; just press the "Next" button. The second page asks you for the location of the new file (in the figure above, "tc_5_01"), and a file name (above, "cubesat_mission").

Upon pressing "Finish", a new mission file is created. It has the file ending ".mission", which is not shown in the ESABASE2 Explorer, and does not need to be specified in the wizard.

A mission editor is automatically opened on the newly created file.

3.1.2 Mission Editor

If the mission editor is not already open, go to the ESABASE2 Explorer view and doubleclick the mission file (recognisable via the planet-plus-orbit icon). A mission editor will be opened, and the Outline and Properties view will be updated.

The following figure shows the application with an opened mission editor.

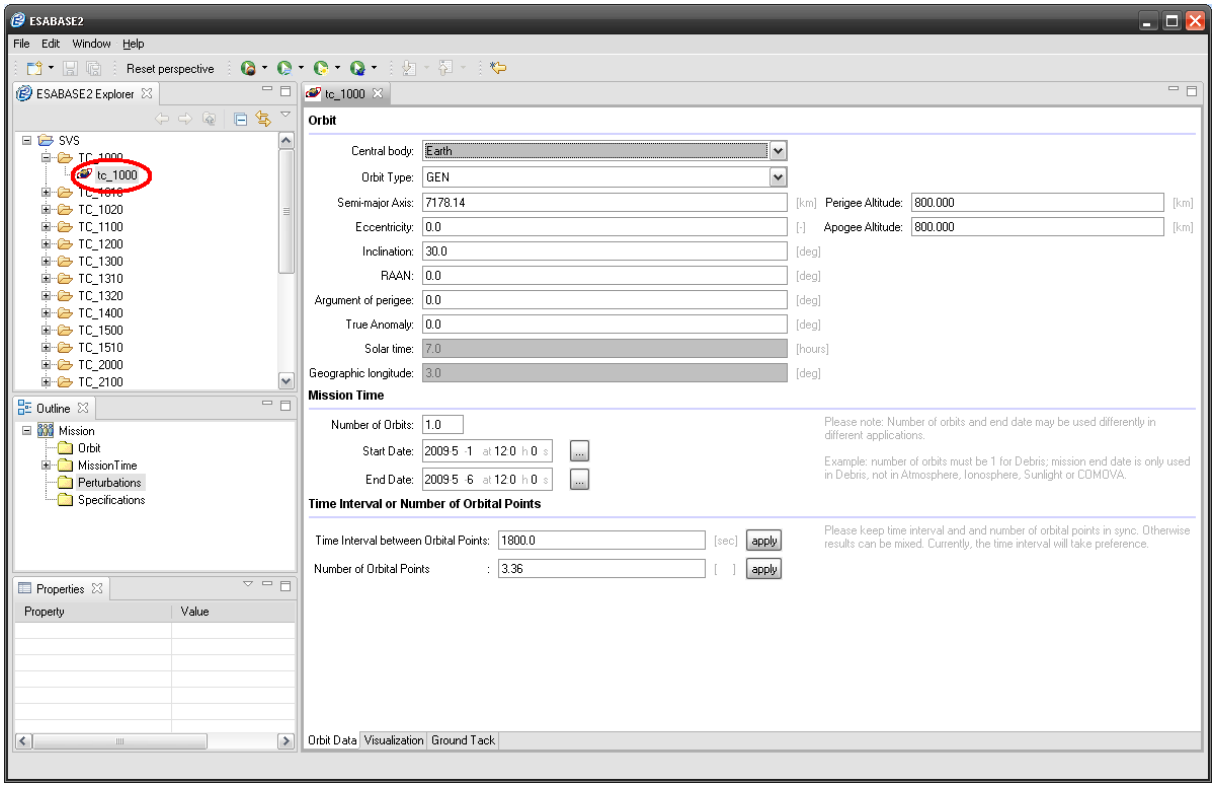

Figure 3-2: Mission Editor, Outline and Properties view

The mission file is circled in red. Below, you see the "Orbit" node selected in the Outline, and further below, the "A" (semi-major axis) and other attributes in the "Properties" view. On the right side, the mission editor itself is shown.

The mission editor contains all necessary parameters to specify an orbit and the number of orbital points to be used during an analysis. It has three sections:

- Orbit
- Mission Time
- Time Interval or Number of Orbital Points

Starting with the "Orbit" section, you can choose whether the central body for the mission should be “Earth” or “Moon”. Please note that “Moon” is only an option for Debris analysis.

When choosing “Earth”, you can choose between a predefined geostationary ("GEO"), a predefined sun-synchronous ("SUN") or a custom ("GEN") orbit in the first combobox of the editor.

In most cases, you will probably want a custom orbit; for that, specify the Keplerian elements in the text fields below the first combobox.

- semi-major axis [km]
- eccentricity]0..1]
- inclination [deg]
- right ascension of the ascending node [deg]
- argument of perigee [deg]
- true anomaly [deg]

Semi-major axis (A) and eccentricity (E) can alternatively be defined using apogee altitude and perigee altitude (textfields to the right). Whenever you leave ("unfocus") one of the latter text fields, A and E will be re-calculated automatically.

Note that for analysis purposes, A and E are always the values that will be evaluated; apogee and perigee altitudes are only helper input methods.

In the next section ("Mission Time"), you specify start and end date of the mission, and the number of orbits. Please note that some solvers (e.g. Debris) will always use number of orbits = 1, ignoring deviant input; this is hardcoded into the Debris solver and cannot be changed by ESABASE2.

The last section, "Time Interval or Number of Orbital Points", is used to specify how many orbital points you want. A higher number of orbital points will increase the accuracy of analysis results, but take longer to compute.

Whenever you make changes in one of the text fields, press the "Apply" button next to it. If time interval and number of orbital points are not in sync, results may vary depending on the solver you use.

Please note that the number of orbital points defined in the mission file may differ from the number of actual orbital points calculated by the orbit generator during the analysis. This can happen because the mission editor does not consider perturbations (see 3.1.3) which lead to acceleration or deceleration of the spacecraft.

Changes on the data in the textfields of the “Orbit Data” tab are directly sent to the visualisation tabs, called “Visualization” tab (3D-view) and “Ground Track” tab (2D-view). If you enter data outside the specified interval an error popup window is shown. Additionally the

textfield background containing the wrong data is coloured red. The visualisation tabs are not updated with wrong data inputs.

3.1.3 Mission Outline and Properties View

Expert users of the original ESABASE may edit further properties using Outline and Properties view. The following figure shows the pertinent points in the GUI.

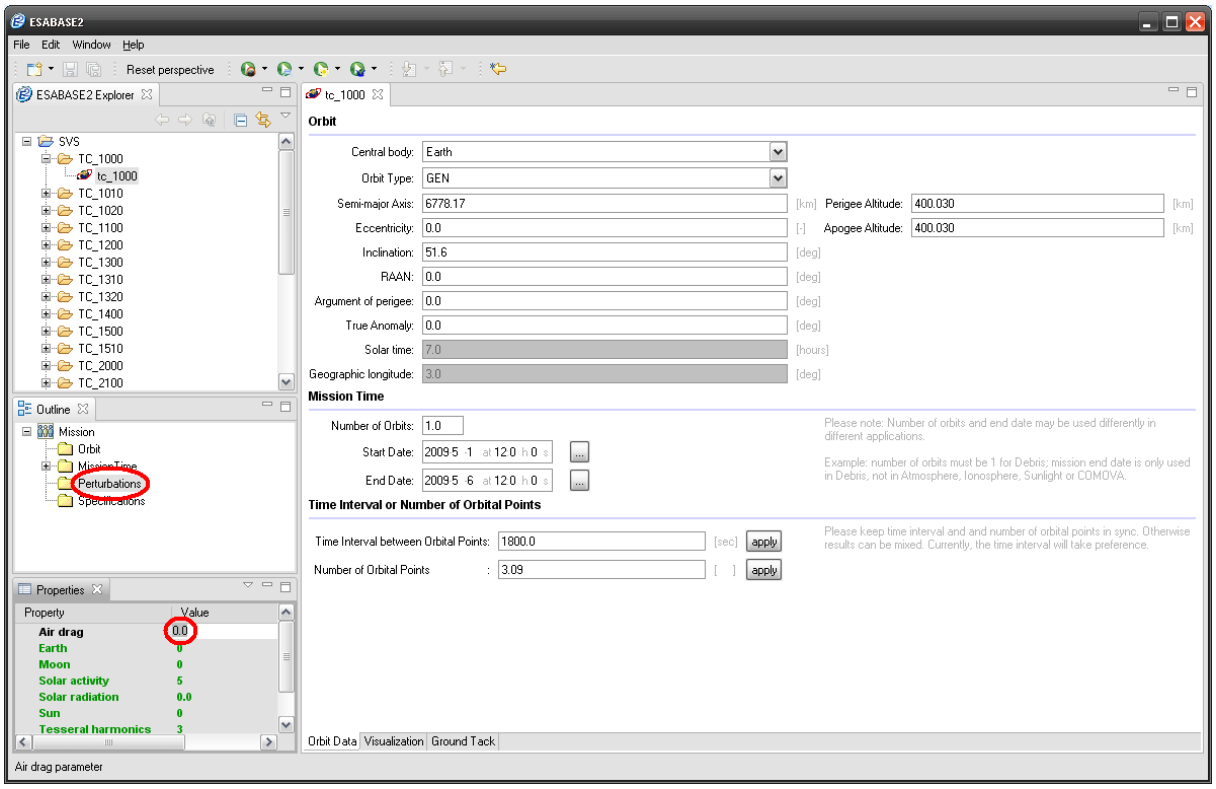

Figure 3-3: Mission Editor, with Pertubations in Outline.

Select the "Perturbations" node in the Outline and click on a value in the Properties view to edit it (if you cannot see an attribute, click "Window → Preferences" in the menubar and set your "user mode" to "expert").

- Wibair — air drag parameter
- Wibspr — solar radiation pressure parameter
- Sun — flag to include the gravitational perturbations caused by the sun
- Moon — flag to include the gravitational perturbations caused by the moon
- Earth — flag to include the gravitational perturbations caused by the Earth
- Kzonal — number of zonal harmonics of the earth's gravity field,
- Ktess — number of tesseral harmonics of the earth's gravity field
- Ncira — number of the CIRA (COSPAR International Reference Atmosphere, 1965) atmospheric density model (0=very low solar activity, 10=extremely high solar activity).

3.1.4 Mission Visualisation

To visually verify the correctness of the specified orbit, you can click the "Visualisation" tab at the bottom of the mission editor, as shown in the following figure.

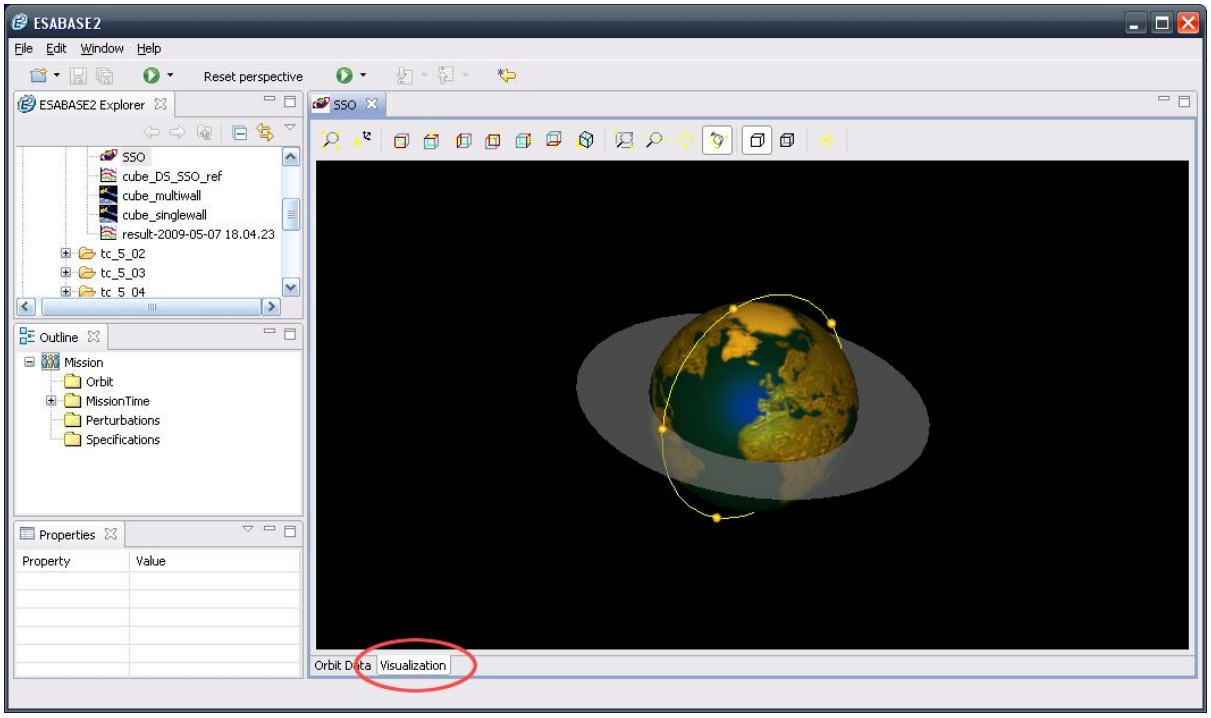

Figure 3-4: Mission Editor, Visualisation page

The "Visualisation" tab is circled in red. In the 3D view, you see the orbit as a yellow circle, with tiny globes at the approximate spots of the orbital points. The equatorial plane is shown in faint white.

At the top of the editor, you see a toolbar for manipulating the viewing perspective (scrolling and zooming). The buttons are the same as in the geometry editor, and will be handled in chapter 3.2.

Please note that this view is not available when choosing “Moon” as central body in the “Orbit Data” tab.

3.1.5 Ground Track

The "Ground Track" view is second possibility to visualize the mission data, set in the "Orbit Data" tab.

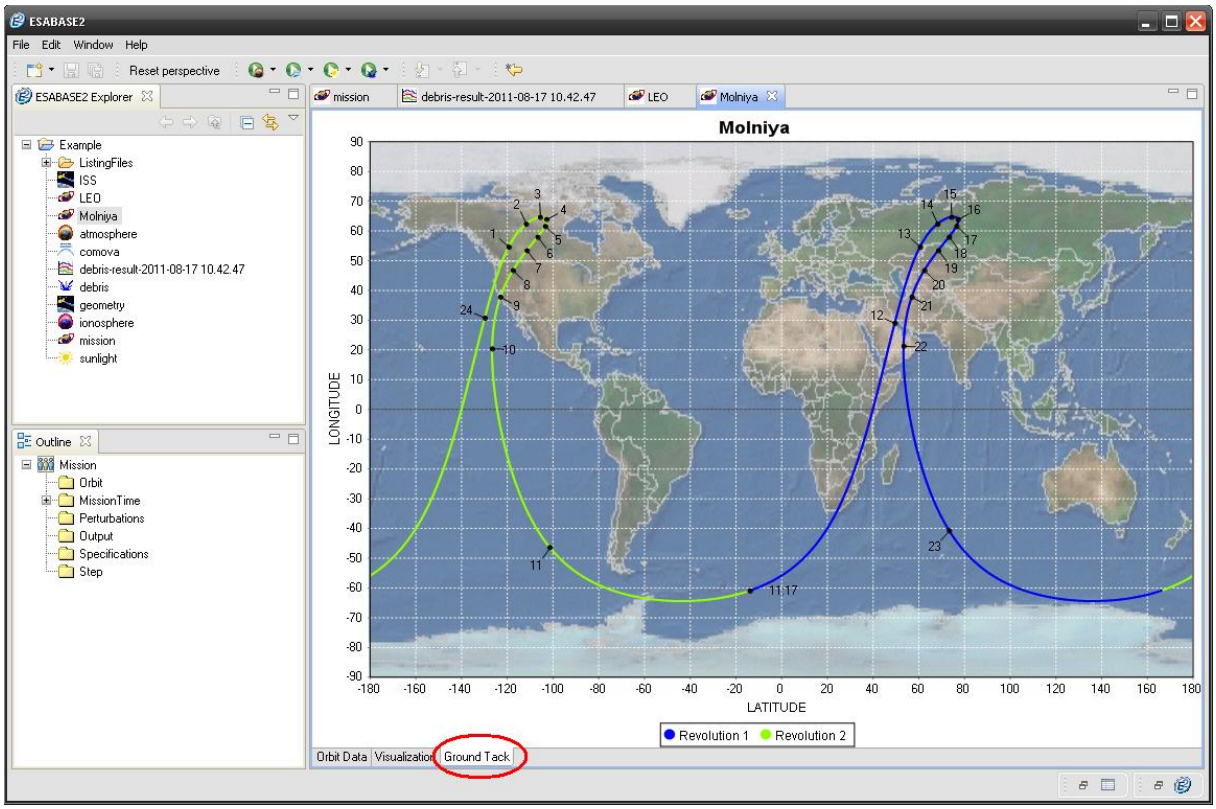

Figure 3-5: Mission Editor, Ground track page

The "Ground Track" tab is circled in red. In the 2D view, you see the orbit as a ground track, coloured in green and blue. Different colours stand for different revolutions as shown in the legend. In the background you can choose between pictures of the earth or shore lines. The equator is highlighted in light grey. Near the orbit hourly time steps are placed.

On right mouse click you can open a popup menu to change for example the title or the axis range of the chart. Further options are the switch the legend, time steps or gridlines on or off.

On left mouse click you can zoom in the chart. To rebuild to initial view there is an option in the popup menu.

You have also the option to save the chart as an image in the ".png" file format.

Please note that this view is not available when choosing "Moon" as central body in the "Orbit Data" tab.

3.1.6 Import of INP Files

Users of the original ESABASE can re-use their mission specifications by importing .INP* files. Select "File → New → Mission" in the menubar and the wizard shown in the following figure will appear.

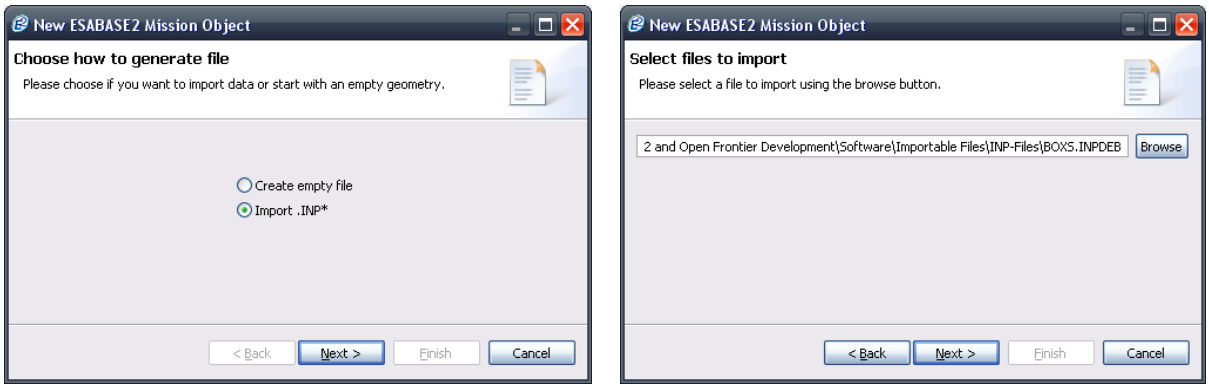
New ESABASE2 Mission Object
Choose how to generate file
Please choose if you want to import data or start with an empty geometry.
Create empty file
Import .INP*
< Back
Next >
Finish
Cancel
New ESABASE2 Mission Object
Select files to import
Please select a file to import using the browse button.
2 and Open Frontier Development\Software\Importable Files\INP-Files\BOXS.INPDEB
Browse
< Back
Next >
Finish
Cancel

Figure 3-6: Mission file creation wizard, for INP files

On the first page, "Create empty file" is the default option. Change this to "Import .INP*" and click "Next". Then, on the second page, enter the location of the INP file and press "Next" again. On the last page, enter file name and location within the project.

Upon pressing "Finish", the file will be created and the mission editor for it will be opened.

3.1.7 Trajectory File

As an alternative to a mission file, it is also possible to use so-called trajectory files to define a set of state vectors to define an orbit.

A trajectory file is an ASCII based text file with the ending `trajectory`. An example file is displayed in Figure 3-7.

```
*****************************************************************
* Example trajectory file.
*
* Head defines the coordinate system and the epoch format:
*  Possible origin bodies:  (currently)
*     Earth
*     Origin is optional, if it is missing, Earth is assumed to be the origin body
*
*  Possible target bodies:  (currently)
*     Earth
*     Moon
*
*  Possible coordinate system:
*     ECI: Earth Centered Inertial (EME2000)
*     LCI: Lunar Centered Inertial (Moon_J2000) Mean pole (z),
*                                  IAU node of J2000.0 (x) System
*  Possible epoch format:
*     MJD: Modified Julian Day 1950
*     UTC: CCSDS ASCII time code A: YYYY-MM-DDThh:mm:ss.d->d
*
* Body provides the state vectors with appropriate epoch
*  Format: epoch X Y Z X_dot Y_dot Z_dot
*****************************************************************
Origin: Earth
Target: Earth
Coordinates: ECI
Epoch: MJD
*****************************************************************
$$BEGIN
23345.00000000    6778.140000000000   0.000000000000E+00   0.000000000000E+00   0.000000000000E+00       4.763311364744       6.009803255004
23345.00803472    4790.555931407092    2976.730920346020    3754.522959103103      -5.427550598326       3.367064289884       4.243334186176
23345.01606944      -4.957103210008    4209.151451932076    5302.945373229049      -7.670596428697  -0.922875698557E-03      -0.014763091198
23345.02410417   -4796.570230341734    2974.083141187338    3734.181735055230      -5.414156716544      -3.374132441261      -4.271484386698
23345.03213889   -6766.522140421070      -8.814993674923     -35.128124323782       0.037419559084      -4.771399704377      -6.019872793309
23345.04017361   -4749.872465879867   -2986.537000237665   -3783.717300211380       5.466768768839      -3.360031482636      -4.214958005963
23345.04820833      60.790730335324   -4209.238301408956   -5302.686766193330       7.670112099350       0.018709646814       0.064632753908
23345.05624306    4836.782926193289   -2964.606413006576   -3704.808445679707       5.374732432089       3.380857829556       4.299068333662
23345.06427778    6778.140000000000   0.000000000000E+00   0.000000000000E+00   0.000000000000E+00       4.763311364744       6.009803255004
$$END
```

Figure 3-7: Trajectory File

You can create this file via the context menu of a project or folder within a project: New -> Other -> File, then press "Next" and enter a name for the file (e. g. `transfer.trajectory`). Please note that the file must have the appropriate extension, otherwise it will not be recognized by the Analysis Wizard and cannot be chosen for an analysis.

It is also possible to create and edit the file with the text editor you prefer, to save the file to disk with this editor, and use drag and drop to move it into the folder of your ESABASE2 project in the ESABASE2 explorer.

Certain syntax must be matched in order to allow ESABASE2 to use the file for an analysis:

- Origin: The origin defines the central body from which the mission specified by the trajectory file originates. It can be omitted, and if you do this, the default value "Earth" is used internally. The second valid option is to define it, and to use either "Earth" or "Moon".
- Target: Like origin, "Earth" and "Moon" are viable options, but unlike the origin parameter, this parameter has to be specified.
- Coordinates: It is necessary to define the coordinate system used for the supplied state vectors. "ECI" and "LCI" can be chosen.

- Epoch: This option must also be specified. It defines the epoch format, and valid options are "MJD" and "UTC".
- For each supplied state vector, one line shall be used.
- The section in which the state vectors are defined shall be started with the $$BEGIN tag, and should end with $$END.
- The format of the state vectors is "`epoch X Y Z X_dot Y_dot Z_dot`".

Please note that trajectory files can only be used for the Debris application. This mechanism does not work for any other application. For more details on how to use the trajectory file, see /2/.

3.1.8 Orbit Propagator

This subsection describes details of how orbital points are calculated from the mission specification. It is particularly interesting to original ESABASE users. The subsection is structured as follows:

- *Introduction to SAPRE*: Description of the SAPRE orbit propagator.
- *Restrictions of SAPRE*: Explains limitations of SAPRE.
- *Input Parameters for Sapre*: Describes how SAPRE views the input parameters in the mission editor.

3.1.8.1 Introduction to SAPRE

Many ESABASE applications, and the pointing facility, need to know the position of a spacecraft on its orbit at successive times during an analysis. For the purpose of generating this information in a standard form, an orbit generator is provided with ESABASE2. The orbit generator currently used is SAPRE.

The SAPRE orbit generator uses a 4^{th} order Runge-Kutta routine with fixed step size to integrate the equations of motion, expressed in terms of osculating orbital elements. It is a general purpose orbit generator, and may be used for close earth orbits, geostationary orbits, or highly eccentric orbits. It was also extended to allow the use for lunar orbits. Facilities exist for taking into account the first few harmonics of the earth's respective Moon's gravitational field, perturbations due to the Sun's and Moon's (for Earth) respective Earth's (for Moon) gravity fields, air drag (for Earth using the CIRA atmospheric density model 1965) and solar radiation pressure (for Earth).

The input to SAPRE is the orbit specified via the mission editor. The orbital elements are specified in a standard format, and control parameters and details of perturbing influences to be modelled can be specified in the respective "Outline" window (see Figure 3-3). The output is saved for each orbital point in the result file and gives the position and velocity of the spacecraft at specified intervals, and also upon eclipse entry and exit. This information is read by the applications requiring orbital information. SAPRE also generates an ASCII report file which can be displayed on the result file "Listings" page, but is also available in the "ListingFiles" directory of the active ESABASE2 project.

It should be noted that SAPRE is intended for the generation of positions over a fairly small number of orbits for use in ESABASE analyses. It is not suitable for analysing the long-term behaviour of the orbit itself. The restrictions on the program are discussed in greater detail in section 3.1.8.2 below.

3.1.8.2 Restrictions of SAPRE

There are a number of restrictions which should be kept in mind when using the orbit generator. SAPRE does not correctly handle orbital inclinations approaching zero. For this reason a minimum inclination of 1E-3 degrees is enforced on input. If the inclination falls below this value at any stage, the program may fail. Thus an orbit exactly in the equatorial plane cannot be modelled. This is most likely to affect a geostationary orbit, where the above restriction requires the orbit to oscillate at least 700 m from the equatorial plane.

The user may include the effects of harmonics of the earth's gravitational potential by specifying how many zonal and tesseral harmonics to include (see KZONAL and KTESS below). However sectoral harmonics are not included and may not be modelled. These terms (with the symmetry of the segments of an orange) are often not important, as during one complete orbit around the earth the effects average to zero. Thus in general only small perturbations result with period no greater than one orbit. But for a geosynchronous orbit, which stays within a limited range of geographic longitudes, this is not the case. In fact the first sectoral harmonic (J22 and K22), corresponding to the ellipticity of the equator, is the major perturbing influence for a geostationary satellite, producing an oscillation about two stable points in the equatorial plane with a period greater than 780 days. SAPRE entirely neglects this effect. However for runs of no more than a few days, the results will be a fair approximation. There is no sectoral harmonics restriction for the lunar orbits.

3.1.8.3 Input Parameters for SAPRE

The following table provides an overview of the orbital parameters used to specify the orbit of the spacecraft, as seen by the SAPRE orbit propagator.

Parameter	Type	Range	Description
hapo	real	≥ 0	altitude of apogee [km]. **Note:** altitude of apogee = apogee radius minus the celestial body's radius Re (Earth: 6378.165 km, Moon: 1738 km).
hper	real	≥ 0	altitude of perigee [km]. **Note:** altitude of perigee = perigee radius minus the celestial body's radius Re (Earth: 6378.165 km, Moon: 1738 km).
A	real	> 6378.165	semi-major axis [km].
E	real	0 ≥ e > 1	eccentricity]0..1].
incl	real	0.001 ≤ INCL < 180	orbital inclination [deg].
raan	real	0 ≤ RAAN < 360	right ascension of the ascending node [deg].
argper	real	0 ≤ argper < 360	argument of perigee [deg]
trano	real	0 ≤ trano < 360	true anomaly at the mission start epoch [deg]

Table 3.1: Orbit input parameters

Please note that the specification of the perigee and apogee altitudes, or alternatively of the semi-major axis and the eccentricity, can be used. The input of any of the values results in a re-calculation of the other values. If the resulting orbit is not valid (e.g. due to a perigee altitude below the Earth's surface), the respective input field will be marked in red.

The parameters of the solar radiation pressure and the atmospheric drag can be modified in the respective "Outline" to consider the corresponding perturbations. Please bear the following definitions of the both parameters in mind. These definitions are provided in /11/.

The solar radiation pressure parameter is defined as:

$$Solar\_Radiation = 0.451 \cdot 10^{-8} \cdot K \cdot \frac{A}{M}$$

where *K* is the material parameter:

K = 1 – *gamma* + *rho* plate

$K = 1 - gamma$	sphere
$K = 2$	plate with perfect specular reflection
$K = 1$	sphere with perfect specular reflection or perfect absorption
$K = 1.44$	sphere with perfect diffuse reflection

rho and *gamma* are the reflectivity and the transitivity of the satellite.

A [m²] is the total reflectivity area of the object, for the spherical satellites it is the cross sectional area. *M* [kg] is the mass of the satellite.

The atmospheric drag parameter is defined as:

$$Atmospheric\_Drag = 0.5 \cdot 10^6 \cdot c_D \cdot \frac{A}{M}$$

Where c_D [-] is the drag coefficient, *A* [m²] is the cross sectional area of the satellite and *M* [kg] is the mass of the satellite.

3.2 Geometry

A spacecraft geometry model is prerequisite for a full analysis. This section describes how you can build a geometry in a CAD way, using the geometry editor. It is structured into the following subsections:

- *Geometry File*: How to create a new geometry.
- *Geometry Viewing*: How to navigate around the geometry editor.
- *Geometry Editing*: How to add and modify geometrical elements.
- *Geometry Outline and Properties View*: Describes the tree structure of the geometrical model as an alternative view on the data.
- *Boolean Operations*: How to perform Boolean Operations.
- *STEP and BAS File Import*: How to import files from other CAD programs and from the original ESABASE.
- *GDML Export*: How to export geometry structures into GDML format.

We start with the geometry file creation.

3.2.1 Geometry File

A default geometry file has been created together with the project. If you want to add further geometry files, choose "File → New → Geometry" from the menubar; a wizard as shown in the following figure will open.

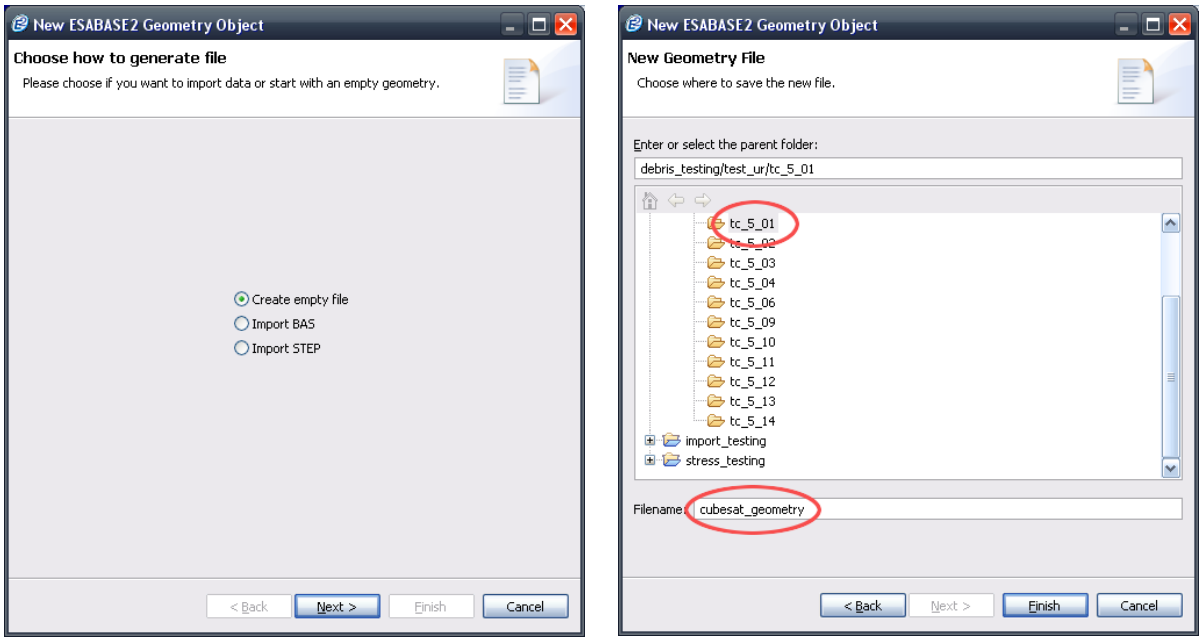

Figure 3-8: Geometry file creation wizard

On the first wizard page, keep "Create empty file" as option and press "Next". The second page asks for the new file's location (above: "tc_5_01") and name (above: "cubesat_geometry", without file ending).

Upon pressing "Finish", the geometry file will be created and the geometry editor opened. Geometry files have the file ending ".geometry", which is neither shown in the ESABASE2 Explorer nor specified by you in the wizard.

3.2.2 Geometry Viewing

If the geometry editor is not yet open, doubleclick on a geometry file in the ESABASE2 explorer. This will open the geometry editor, as shown in the following figure.

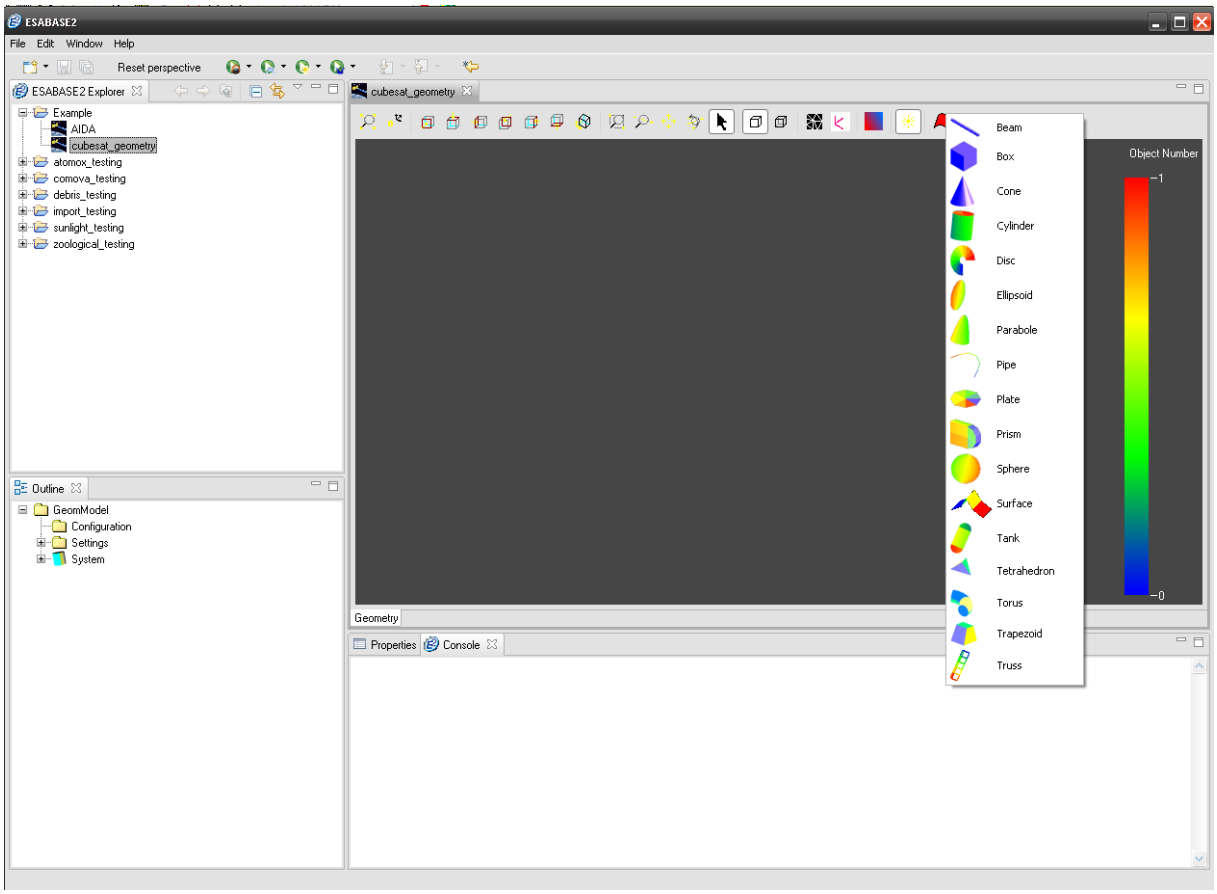

Figure 3-9: Geometry editor, empty file

In the left column, you see the opened geometry file in the ESABASE2 Explorer, the tree structure of the geometry in the Outline (with "System" marked) and the selected node's attributes in the Properties view.

On the right side, you see a blank screen, as there are no shapes created yet. At the top of the editor (directly under the "cubesat_geometry" tab), you see a toolbar to manipulate the geometry.

This subsection's goal is to explain how to view a geometry, not how to edit one (this is the subject of the following subsection). However in order to have something to view, press the button indicated by the red circle in the figure above; this will provide you with a palette of available shapes.

Select the "Box", and in the wizard that opens, just press "Finish". This will produce a box with some sensible default values.

The following figure shows the newly created box in the 3D view, and on the right side a colour scale, which matches colours on the surfaces of the cube to object numbers.

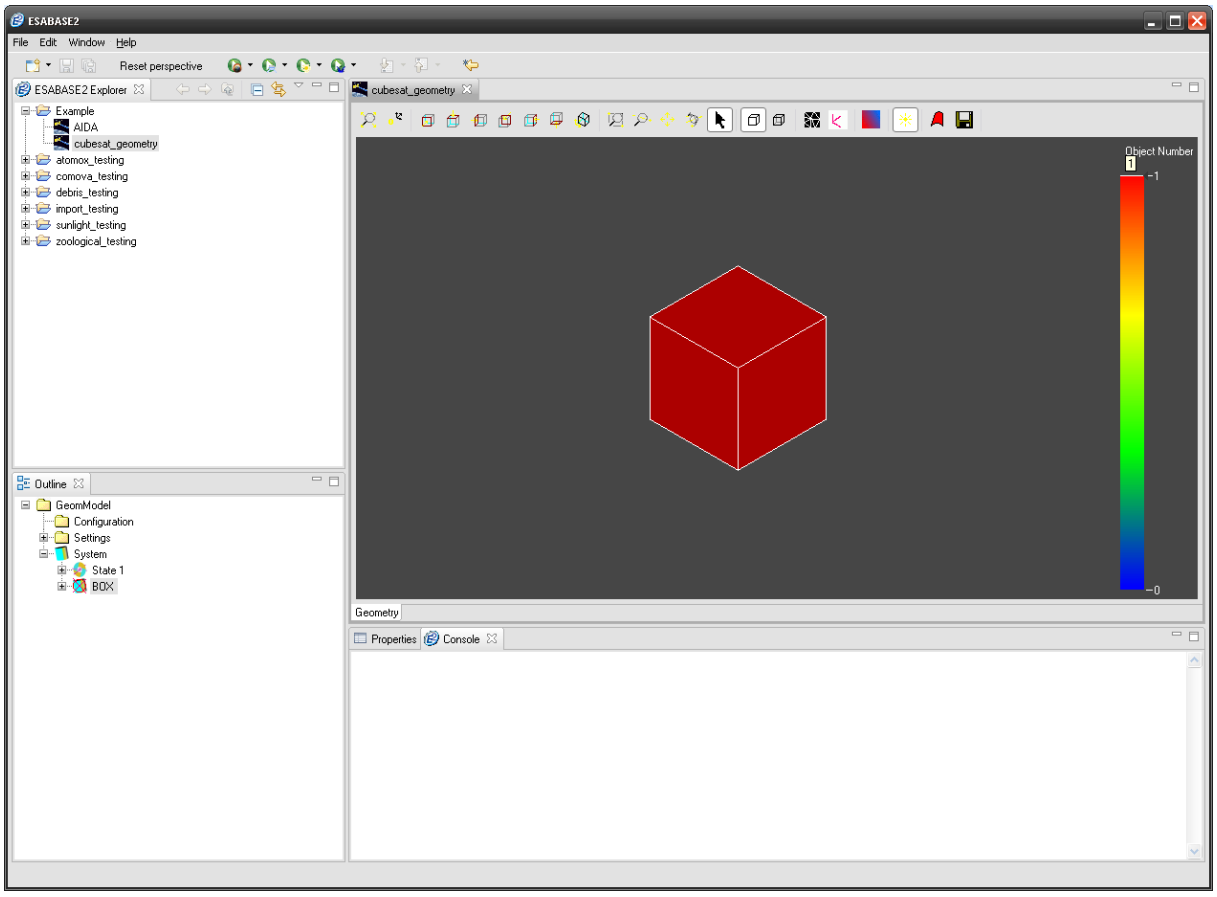

Figure 3-10: Geometry editor, first box

The box can be viewed in two different modes: Unmeshed and meshed. The unmeshed view is the default setting when creating a new geometry. It displays the shape as 3D object. This view is convenient for constructing the spacecraft's geometry, as objects can be manipulated and added very fast. However, to use the geometry with a solver, a mesh needs to be applied to the geometry. Meshing and its parameters are handled in another chapter, but to view the mesh, one needs to toggle the meshed view by using the button depicted below.

Figure 3-11: Geometry button, Toolbar (right side), Mesh button

If this view is chosen for our box, the editor will display the following view:

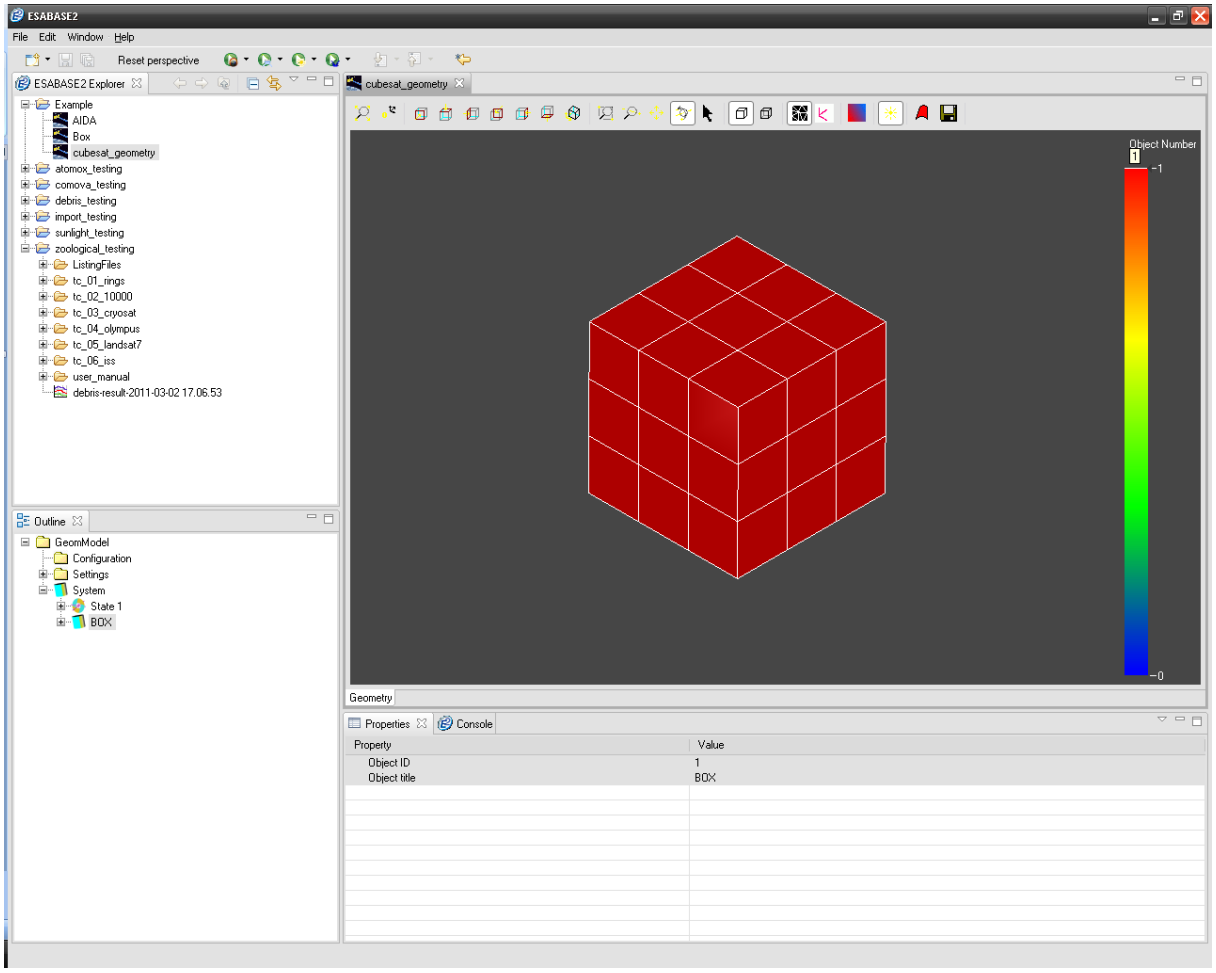

Figure 3-12: Geometry Editor in meshed view

Now the mesher is applied to the shape and elements are displayed. If you look at the outline, you will notice that the box element in the object tree is no longer crossed. This means that a mesh is available for the shape. Only unmeshed objects are marked in the outline with a red cross.

Please note that depending on the complexity of the geometry, switching between unmeshed and meshed view can take some time. Not all shapes are meshed every time the mode is switched. Only those shapes which need to be meshed (if they had no mesh applied yet, if a meshing parameter was changed, if the dimension itself was changed) are meshed. We suggest working in the unmeshed mode to construct the shape and then perform the meshing before using the geometry for an analysis.

Consider now the toolbar, particularly the left part: It concerns manipulation of the viewing perspective. Please refer to the following figure:

Figure 3-13: Geometry editor, Toolbar (left side)

The first two buttons reset the view – the left one tries to fit the geometry to the window, showing as much as possible, while the right one takes a wider perspective, being some distance away.

The next seven buttons allow viewing the model from canonical perspectives:

- front, top, left view
- back, right, bottom view
- axo ("all axis 45°") view

The following four buttons are useful for more interaction than just pressing a button – each toggle button activates a particular interaction mode:

- Zoom area mode allows to select a rectangular region with the mouse that should be zoomed
- Normal zoom mode zoom in and out by dragging the mouse
- Pan mode move the view by dragging it with the mouse
- Rotate mode rotate the object with the mouse

With the first zoom toggle button ( ), the zoom area mode is activated. Leftclick and hold the mouse in the geometry area and drag a rectangle, then release the mouse button. The selected area is now the new viewing perspective.

The second zoom toggle button ( ) works differently: While this mode is active, you can hold the left mouse button and move the mouse to the left or to the right. Left means further away, right means closing in.

Using the third toggle button ( ), you can pan the perspective. Hold the left mouse button and move the perspective with the mouse, then release the mouse button.

The fourth button ( ) is used for rotation. Within this mode, hold the left mouse button and move the mouse cursor to gradually change the rotational viewing angle.

The last button on the left side of the toolbar ( ) is the "Select mode". Being in this mode allows you to click on a shape, which will then be highlighted.

With this, you know how to view a geometry from all sides, and we now come to the CAD editing capabilities of the Geometry Editor.

3.2.3 Geometry Editing

Every geometry consists of a tree structure: Starting with the "System" node, you define a central body and all other pertinent shapes as children of the central body node (or further down the tree structure).

To create the first (and central body) shape, press the "Add object" ( ) button in the geometry editor toolbar, and choose the desired shape. The following shape types are possible:

- *Basic Shapes*: Geometrical primitives which could appear in any CAD program: Box, Cone, Cylinder, Ellipsoid, Ogive, Paraboloid, and Sphere.
- *2D-component Shapes*: Geometries defined partly as a two-dimensional blanket: Disk, Plate, Surface, and Prism.
- *Spacecraft Shapes*: Unusual shapes that aid in the construction of spacecraft: Antenna, Beam, Pipe, Tank, and Truss.

A full description of all shapes can be found in "Annex A: Geometry Shapes". In this section, for the purpose of explaining the geometry editor workflow, we will compose a simple example satellite, as shown in the following figure.

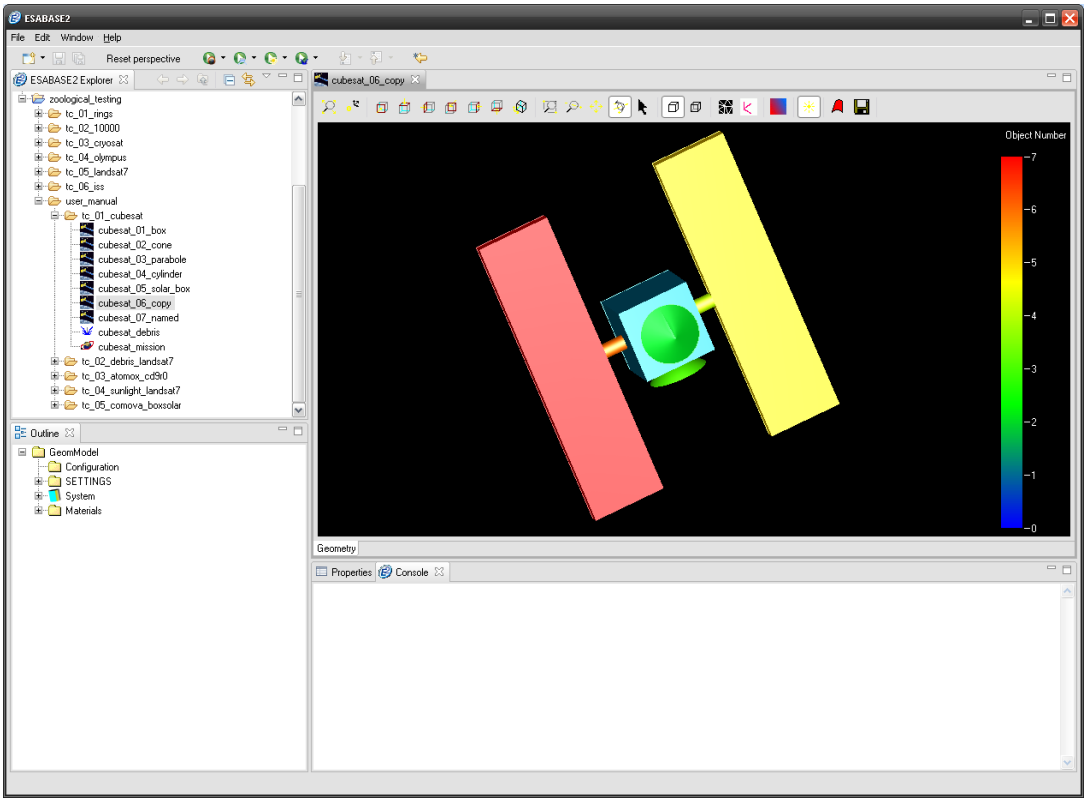

Figure 3-14: Geometry editor, example satellite

The satellite above is a fictional Cubesat derivative, consisting of:

- a box as central body
- a cone as front shield
- a parabole as earth-facing antenna
- two cylinders as solar panel joints
- two flat boxes as solar panels

Re-visit the "Add Object" ( ) button and choose "Box" ( ). A wizard to define the shape will open. Each shape has its own wizard page; other pages are shared among the shapes.

Alternatively, if you already created the box in the previous chapter, select the box in the 3D view (left click while being in selection mode), and then open the context menu (right click), choosing "Modify → Size".

The following wizard pages are available:

- *Size*: This page defines both shape size and point of origin, and is unique for each shape.
- *Meshing*: Every shape is divided into surfaces (flat or curved), which in turn are meshed into elements (always flat, triangle or quad). This page determines meshing detail and active sides.
- *Position and Attitude*: Allows translating and rotating the shape in its own local coordinate system.
- *Kinematic*: Defines joints that allow child elements to move. See subsection 3.3.
- *Pointing*: Defines shape pointing (e.g. solar panel to sun). See subsection 3.4.
- *Material*: Defines the material properties of the shapes surfaces. Only used in some solvers.

Some solvers (e.g. Debris, COMOVA) may introduce additional pages to fulfil their special geometry model requirements. These pages will be explained in the respective solver chapters.

3.2.3.1 Box: Size Page

The "Add Box" command will result in the wizard shown in the following figure.

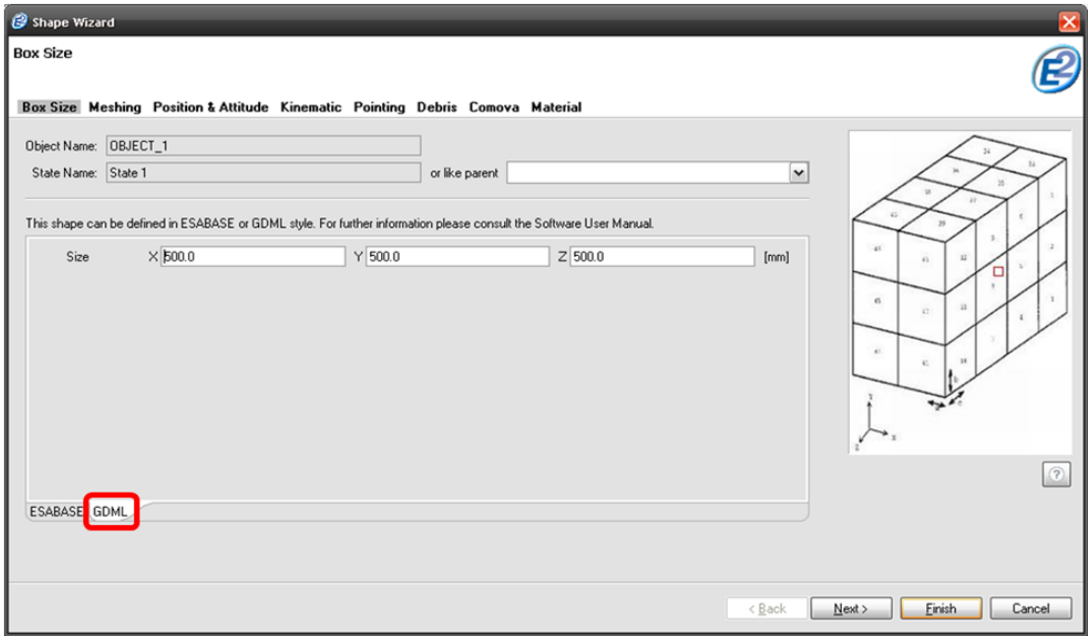

Figure 3-15: Geometry editor, Box wizard, Size page

In the upper section, you can see object name and state name. Usually, you can leave them as is (future versions of ESABASE2 may support multiple states per object).

The box definition itself is quite straightforward. Choose a size in XYZ dimensions, and then specify whether the point of origin is the front lower right corner point or the centre of the box (we recommend the former).

As you can see in the picture (marked by the red box), it is also possible to supply input parameters in GDML style. Besides the box, the wizard offers this functionality for the sphere, the cylinder, the ellipsoid and the cone.

All parameters of the box are described in "Annex A: Geometry Shapes". For our purposes, set 100 mm as side length and press "Next".

If you enter invalid values, the appropriate fields will become red. Please note that for some more complex shapes, certain errors can only be detected after you press "Finish" (these are errors related to surface generation and meshing, which is only done once for performance reasons).

In that case, a popup dialog will inform you, and you have to revisit the wizard in order to correct the values.

3.2.3.2 Box: Meshing Page

The following figure shows the meshing page, which has the same structure for all shapes, but differing default parameters.

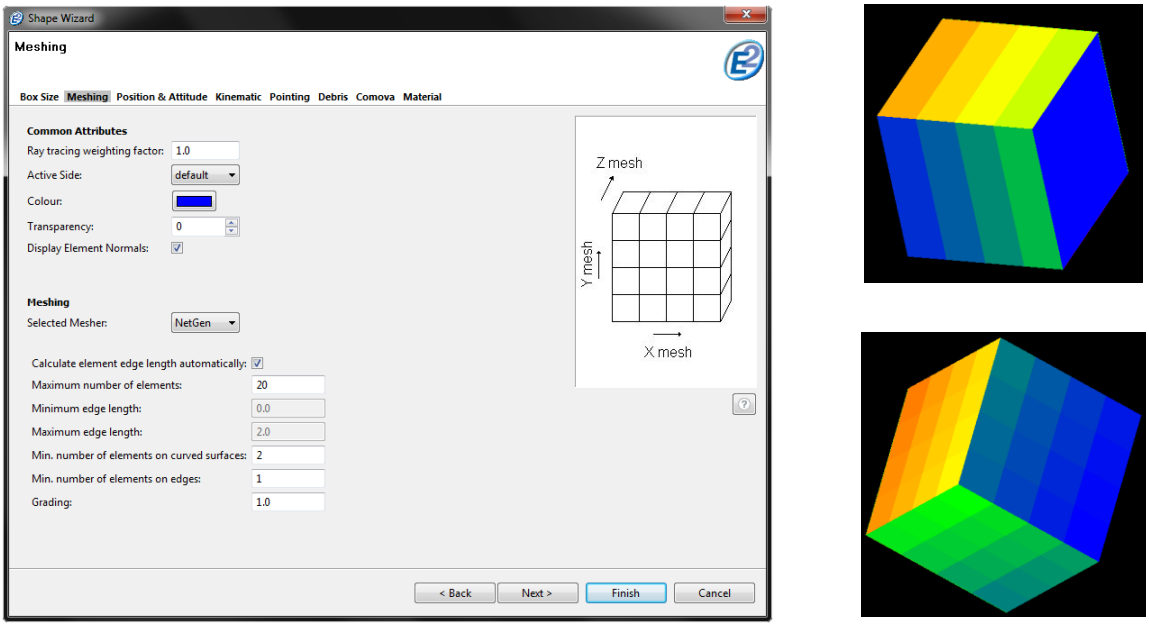

Figure 3-16: Geometry editor, Box wizard, Meshing page

On the meshing page, you can fine-tune the subdivision of surfaces of the object (which may be curved) into meshed elements (which are planar triangles or quads). The normal case is to accept the defaults.

Up to three meshers are available for the shapes: the ESABASE-, NetGen- and OCC-mesher. How many meshers are available for a shape depends on the type of shape. For the shapes from the original ESABASE shape palette, all three meshers are available. For Torus, Tetrahedron and Trapezoid, only the NetGen- and OCC-meshers can be used. The same applies for shapes imported via STEP-import. In the following, the different meshers are explained briefly. For a more deeply understanding of the meshers, please refer to the appropriate user manuals.

ESABASE Mesher

Along any axis, you can modify the number of subdivisions per surface (resulting in distinct "node areas"), and further the number of elements per subdivision. Multiple axis definitions are combined (see figure above): Setting a value to 4 on one axis results in slices on a surface; setting 4 on a second axis results in quad division.

If you just want to fine-tune the meshing, use the second block ("Number of elements"). However, if you use a solver that supports surface nodes (e.g. Sunlight /5/), you can also use the first block ("Subdivision into node areas").

A word on the nomenclature: If you take a surface and e.g. divide it into two (na = 2), there are now 2 node areas. If only one side of the surface is active (see below), there are 2 surface nodes; if both sides are active, there are 4 surface nodes. You can think of surface nodes as being "the active sides of node areas".

OCC Mesher

The OCC mesher works with a value called “deflection”. Deflection is the maximum distance between an edge of your mesh and the corresponding surface. It is measured in mm.

The deflection value can be calculated automatically. If this option is chosen, the deflection will depend on the boundary box around a shape or surface.

The mesher can work in two ways: Either it can mesh the whole shape, or each surface can be meshed individually. The latter option tends to produce more elements than the first.

As a rule of thumb, one can say that the smaller the deflection value is, the finer the mesh becomes.

In some cases the OCC mesher will orientate the mesh into the opposite direction, which e.g. means that the normals of the mesh-elements point inwards the solid. This will confuse the solvers. To avoid this, you may verify the orientation of the relevant elements by enabling the check-box “Display Element Normals”. If the normals point into the wrong direction, enable or disable the check-box “Invert Element Normals”.

NetGen Mesher

For the NetGen mesher, you can choose between two basic modes: One based on maximum numbers of elements, and one based on edge length specifications (change the top check-box to activate the appropriate fields). For a full discussion of the available parameters, refer to the Netgen user manual /9/.

Further down the meshing page (and shown in the figure below), the "Active Side" setting determines which sides of the shape's surfaces are affected by the solvers (mostly via ray-tracing).

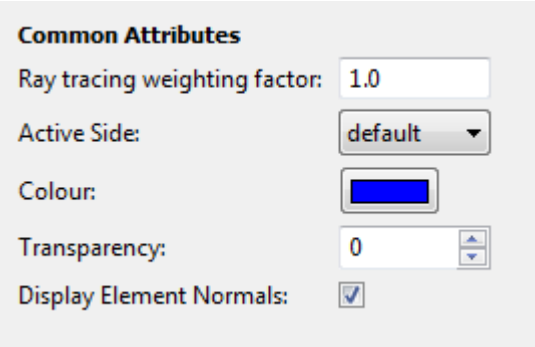

Figure 3-17: Geometry editor, Box wizard, Meshing page, Raytracing and sides

For volume shapes, there are outer (positive) and inner (negative) sides; for surfacic shapes (e.g. plate), one side is usually declared positive, the other one negative. In the meshing page, the following active sides choices are possible:

- positive/negative: only the outer/inner (or front/back) side of a shape is affected.
- both/none: both or no sides are affected.
- default: Depends on the shape. For volume shapes, the default is "positive", as the inner side (e.g. of a box) is supposed to be never reached by raytracing. For surfacic shapes, the default is "both".

In addition, you can define a ray tracing weighting factor; it defaults to 1.0 and sets how many rays are emitted from the surfaces (e.g. setting it to 2.0 means that the surfaces of this shape emit twice as many rays as any other shape in the geometry).

Please note that solvers may or may not respect the ray tracing weighting factor.

For each object, a transparency can be defined. Transparency reaches from 0 to 10. 0 means that the object is solid, where at level 10 it becomes completely transparent.

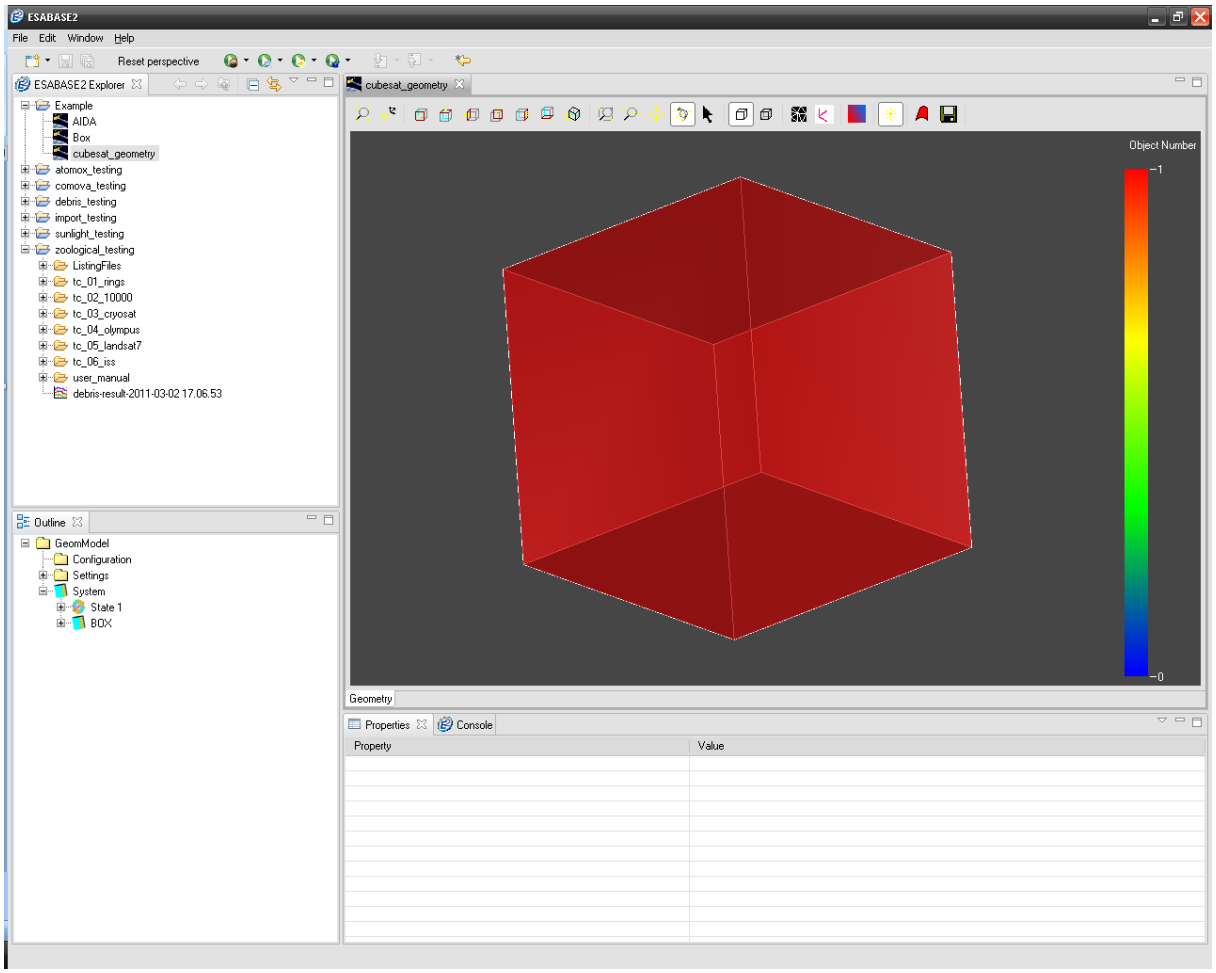

Figure 3-18: Geometry editor, transparency set to 5

Moreover, you can choose the color of the shape. The default is to color it according to object number, but you can override it with a fixed color here.

Finally, you may check the orientation of the mesh elements by enabling “Display Element Normals”. This will display a perpendicular arrow at the centre of each mesh element.

Close the Box wizard now by pressing "Finish", as we will now continue with a cone as a child to the box.

3.2.3.3 Cone: Position and Attitude Page

Each shape has its local coordinate system, which is initially set to (0, 0, 0) within its parent's local coordinate system. The Position and Attitude page allows you to arrange a shape within the coordinate system.

Normally, you define the shape size without regard to placing it (although for some shapes, this would be possible). You then move the shape along the XYZ axis of its local coordinate system ("translation") and rotate it by specifying angles and axis priorities ("rotation").

To further our example satellite, mark the box in the 3D view by leftclicking it, then rightclick to open the context menu and choose "Coordinate Systems → global coordinate system". This will show the global coordinate system.

Rightclick the box again and choose "Add Child → Cone" from the context menu. On the wizards size page, give the cone p1 = (0, 0, 0) and p2 = (0, 30, 0), complement it with diam1 = 80 and press "Finish" to preview the result.

The following figure shows that the cone is somewhat misplaced, especially if seen from the front view (, circled in red):

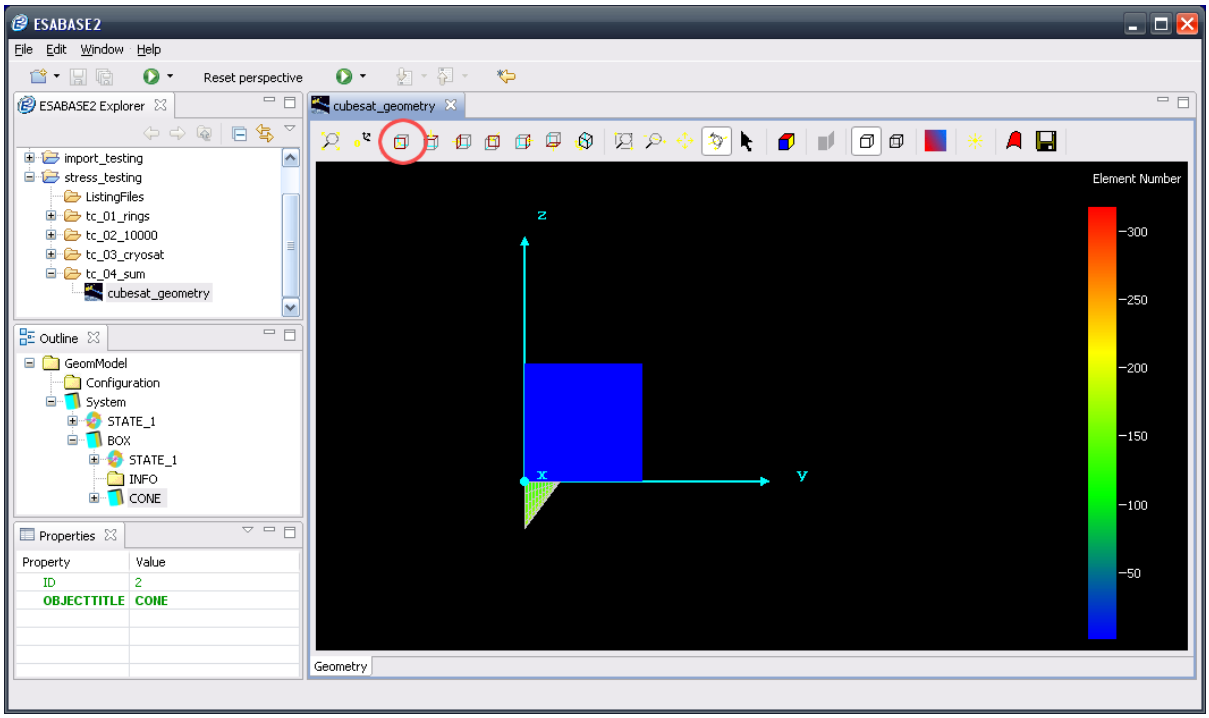

Figure 3-19: Geometry editor, default Cone

To remedy the situation, leftclick the cone to select it, and then choose "Modify → Position and Attitude" from the context menu. This will directly jump to the Position and Attitude page in the Cone wizard, as shown in the following figure.

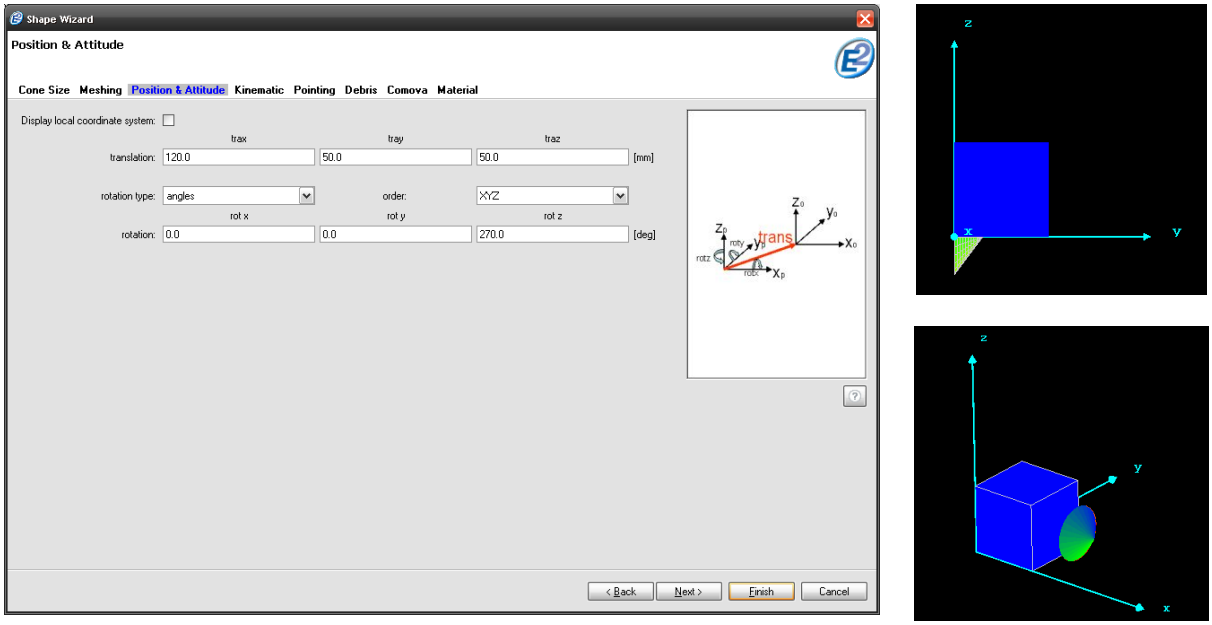

Figure 3-20: Geometry editor, Cone wizard, Position and Attitude page

In the first row, translate the shape by (140, 50, 50) mm. Because X is the velocity direction of the satellite as indicated by the front viewing perspective, this will move the cone to 4cm before the front plate and centre it in Y and Z direction.

In the rotation section below, keep "angles" and a XYZ rotation order, then choose angle 3 (z angle) = 270°. Click "Finish" to view the results, as shown above in the "before" and "after" screen shots.

Please note the checkbox for the local coordinate system. If this is selected, the local coordinate system for that object will be displayed within the editor.

3.2.3.4 Cone: Material Page

We skip the Kinematic and Pointing pages (see subsections 3.3 and 3.4) and concern ourselves with material properties, which always apply to surfaces, not to bodies (i.e. no bodies "filled with aluminium" here). Since we need surfaces, we have to mesh the geometry first.

Press the "mesh" button ( ), so that each object of the geometry will be meshed with the configured mesher and surfaces will be generated.

For our cone, choose "Modify → Cone Size". Although size pages differ from shape to shape, all will contain a "Thickness" parameter of some kind. Choose thickness = 3mm.

Open the material editor page by clicking the "Material" link highlighted with a gray shadow in Figure 3-21:

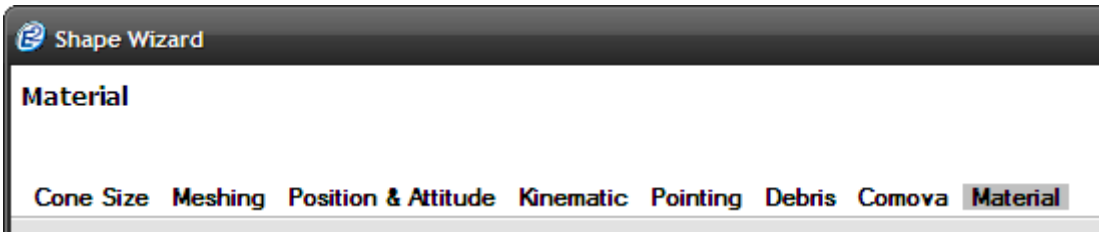

Figure 3-21: Geometry editor, Wizard header

The resulting material overview page will in turn look like the following illustration.

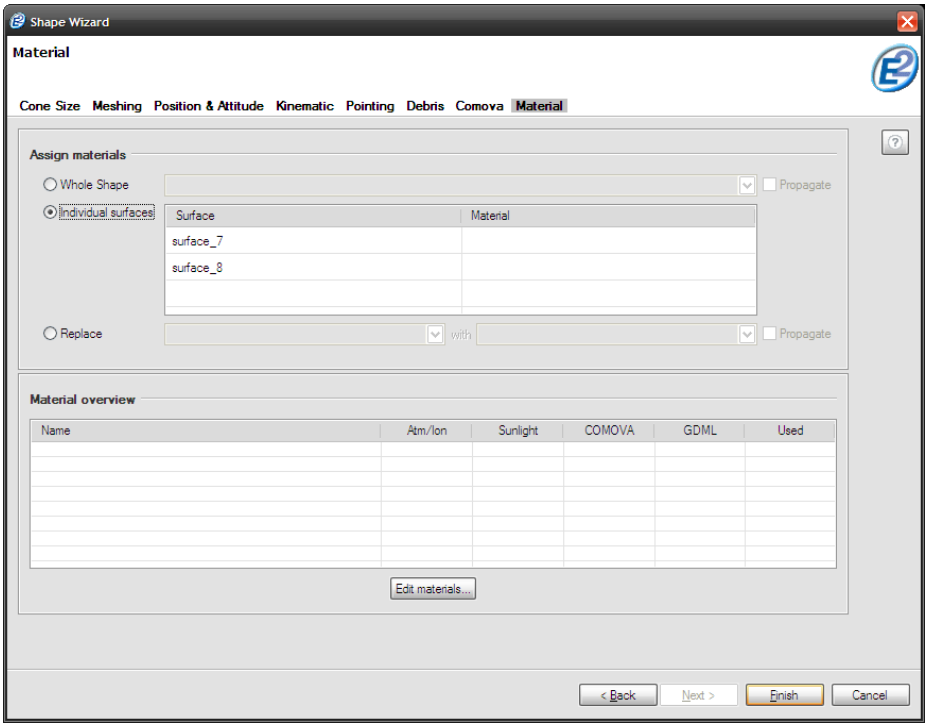

Figure 3-22: Geometry editor, Material page

It is divided into two sections: The upper section named "Assign materials" and below that, the section called "Material overview." At the bottom is a button called "Edit materials…", which opens the Material Editing Page.

3.2.3.4.1 Assign materials section

The upper section allows you to apply materials to surfaces in three ways. You can assign all materials of the current shape the same material by selecting the option "Whole Shape". Or you assign each surface its material individually by using the option "Individual surface". Finally you can replace a material, which is already assigned to one or more surfaces, with another material.

The "Propagate" checkboxes available for the "Whole Shape" and the "Replace" option, allow the fast modification of the current shapes and all its child shapes. By activating the "replicate" checkbox of the "Whole Shape" option, not only will all surfaces of the current shape get a new material but also all surfaces of all shapes that are children of the current shape. For the "Replace" option, it works analogous.

Notice, that currently no material is assigned to any surface. This is the default case.

3.2.3.4.2 Material overview section

The material overview section shows a list of all materials, which are currently stored in the geometry file you are editing. It also shows if the material is useful for certain solvers and if the material is currently assigned to a surface (column "Used"). This is indicated with either an empty or checked box. If you hover over a column title, you can see the requirements a material has to fulfil in order for it to be useful for a solver.

Since we have not added any material to our geometry yet, the list is empty. Adding materials to the geometry can be done on the Material Editing page (see 3.2.3.4.3). Once a material is copied to or created in the geometry it is available in the material combo boxes in the "Assign materials" section.

3.2.3.4.3 Material editing page

We will now use the Material editing page to copy a material from the material library into the geometry file. Open the editing page by clicking "Edit materials…" on the Material Overview page:

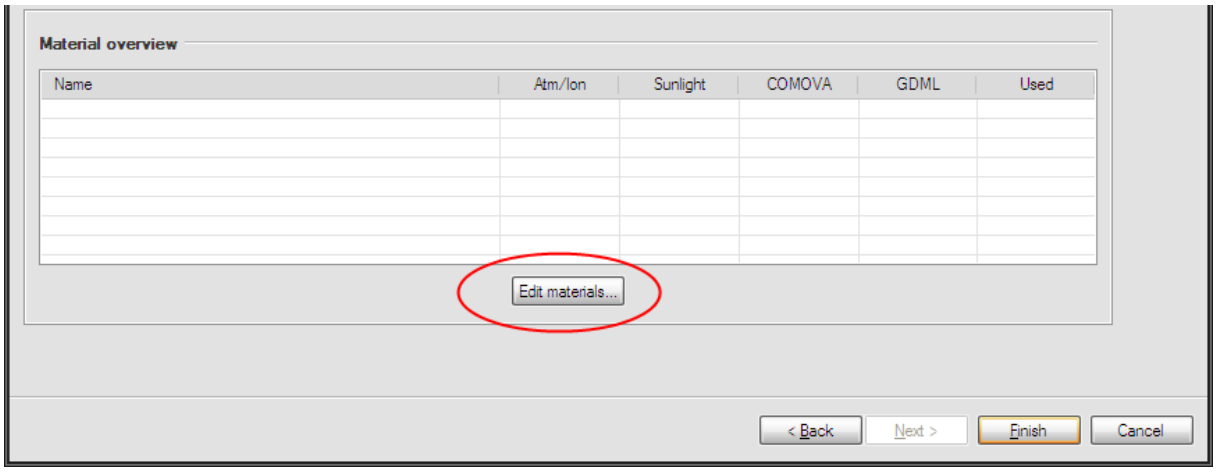

Figure 3-23: Material Overview Page: Edit materials button

The material editing page (Figure 3-24) opens. It is divided into three sections: the geometry materials tree, the library materials tree and the properties view. A toolbar is located at the top.

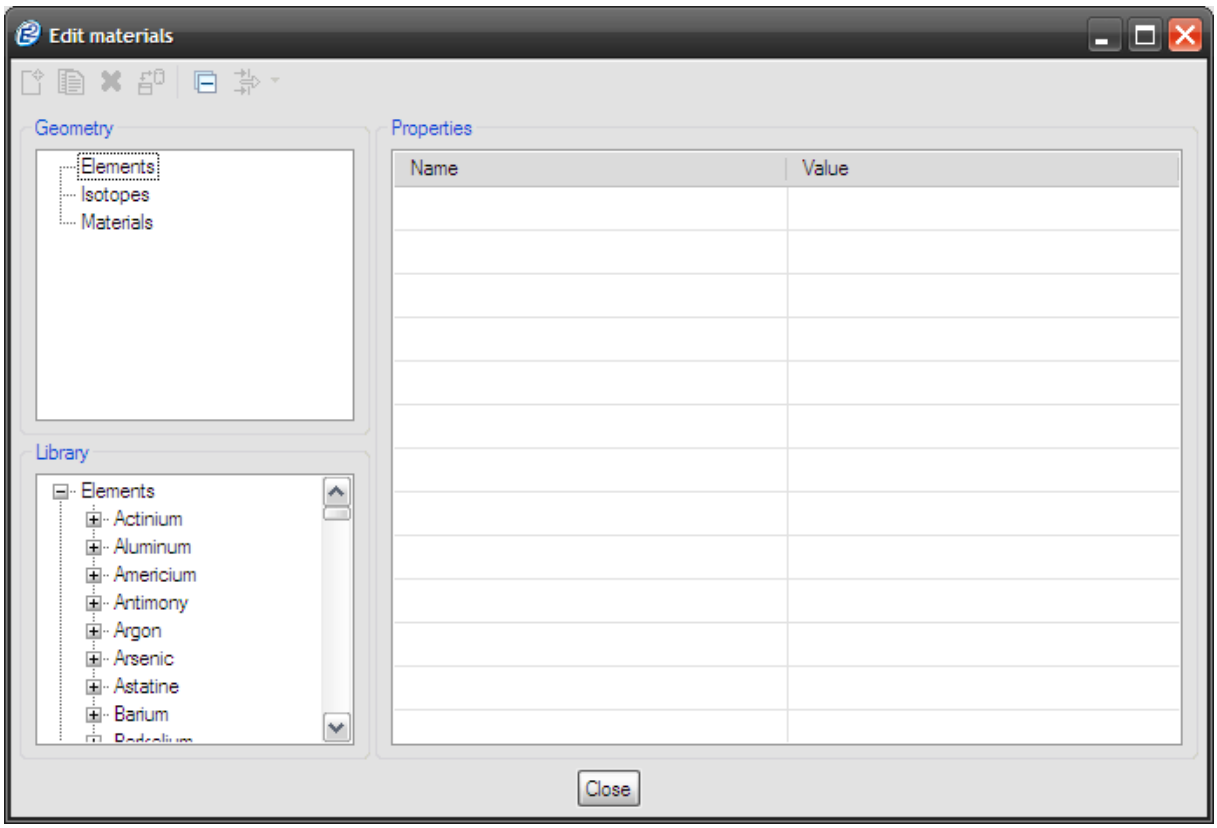

Figure 3-24: Material Editing Page

3.2.3.4.4 Geometry materials

The geometry materials tree lists all elements, isotopes and materials (in the following we call these three types "entities") that are stored in the geometry file. Initially, a geometry file

contains none of these entities. You can either create new entities or copy them from the library.

3.2.3.4.5 Library materials

The library materials tree lists all entities that are stored in the ESABASE2 material library. The material library contains all elements from the periodic table and a selection of the most common materials. Those pre-defined standard materials can never be deleted or changed and work as a basis to create new materials via duplication.

3.2.3.4.6 Properties view

Figure 3-25 shows the properties view populated with the values of the material ALUMINUM(1), which was created by copying the original material ALUMINUM with the "Duplicate" button.

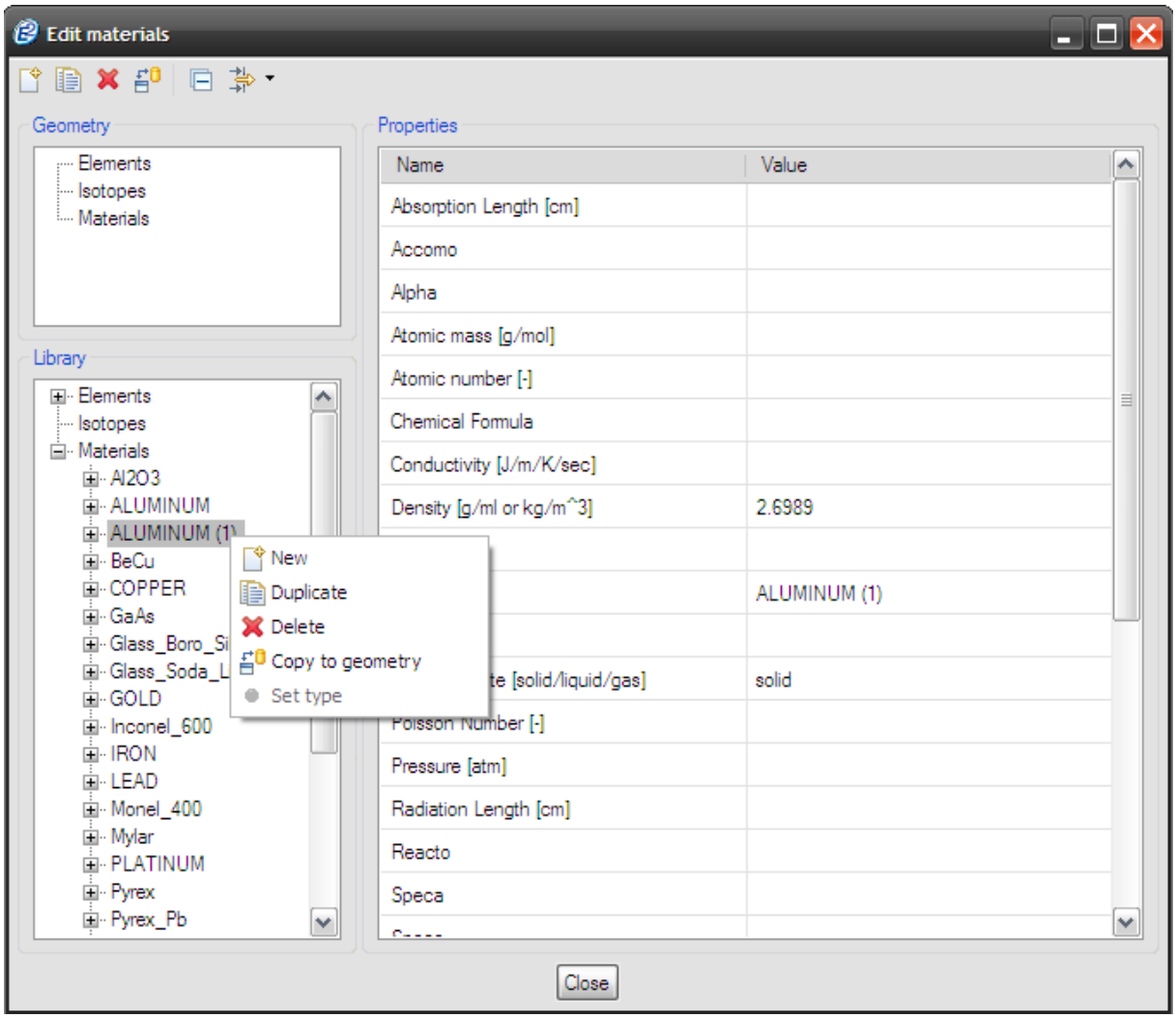

Figure 3-25: Material editing page: Properties view

3.2.3.4.7 Toolbar

The toolbar offers the buttons "New entity", "Duplicate entity", "Delete entity" and "Copy to". All of these actions can also be accessed in the context menu of each tree node. Right-click on a library or geometry tree node to open the context menu.

The toolbar furthermore contains the buttons "Collapse" and "Filter", whereas the context menu has the button "Set type".

All buttons are described below.

Button "New entity"

This button creates a new entity. The type of entity (material, fraction, isotope,…) is determined based on the currently selected tree node. That means, if e.g. the Elements node is selected, the "New" button will create a new empty element. A special characteristic of the "new" button has to do with the creation of composites or fractions. If you create a new composite under a user-defined material or a new fraction under an element or material, it will automatically change the type of the parent element or material to whatever you added to the parent. See description of the button "Set type" below for details.

Button "Duplicate entity"

The "Duplicate" button simply creates a copy of the selected entity and puts it in the same tree (geometry or library tree).

Button "Delete entity"

This button deletes the currently selected entity. It is restricted to entities, which are not referenced by other entities and which are not assigned to surfaces (in the case of materials).

Button "Copy to"

The "Copy to" button allows the user to copy an entity from the ESABASE2 material library into the geometry and vice versa. If an entity with the same name already exists in the target tree, the user has to confirm the overwrite process.

When an entity with child entities gets transferred, the child entities get transferred as well. E.g. a material composed of two elements getting copied results in the two elements getting transferred as well.

Button "Set type"

Materials can have three types: atomic, made of composites or made of fractions. This button enables you to toggle between these types. Initially a new material is considered atomic. When you add a fraction or composite as a child to this material, you have the option to switch the type. Select the Fractions or Composites node and use the "Set material type =" button appropriately.

Button "Collapse"

This button enables you to close the library or geometry tree, whichever is currently selected.

Button "Filter"

If a material is selected, this button allows the filtering of the properties view (see Figure 3-26). Not all properties of a material are useful for a certain solver, e.g. the Atmosphere/Ionosphere solver only uses the properties "Accomo", "Omega", "Reacto" and "Speco". Using the filter, you can easily see which properties are used for each solver.

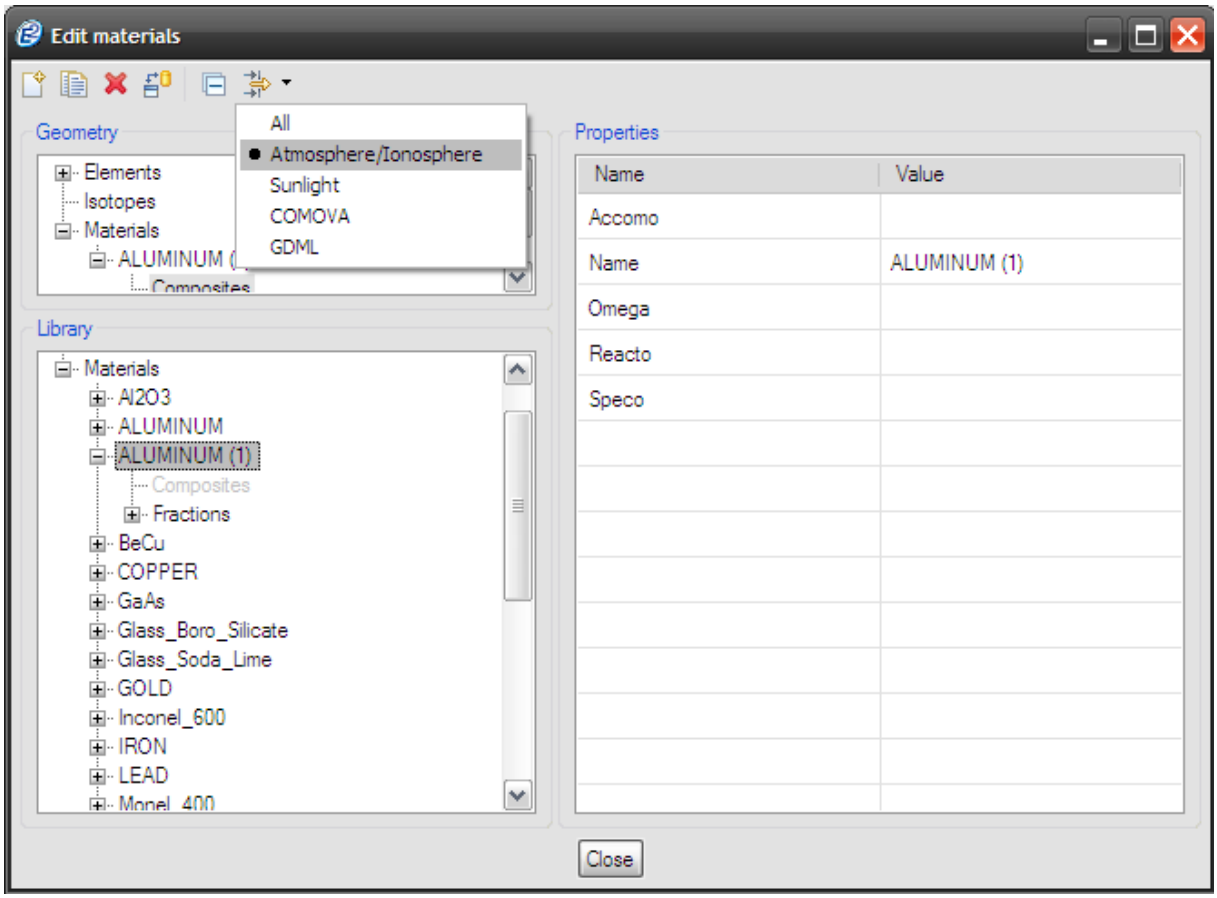

Figure 3-26: Material editing window, Filter button

Now that we know each control element in the Material Editing Window, we can copy the "ALUMINUM" material into our geometry by selection the material and using the "Copy to" button. Notice, that the element "Aluminum" was copied as well, since the material references it as a fraction. When you close the Material Editing Window by clicking "close" at the bottom, you will now see, that the material "ALUMINUM" appears in the Material overview list. As indicated in the list, it is already usable for COMOVA analyses and for GDML export.

To assign the material to all of the cone's surfaces, select the "Whole Shape" option in the "Assign materials" section and select the material in the combo box (see Figure 3-27). Pressing "Finish" will result in the storage of the material in the geometry file and the final assignment of the material to the surfaces.

If you re-open the material editor for the Cone again, you will see, that each surface has the material "ALUMINUM". Furthermore, the material overview shows, that ALUMINUM is now being used.

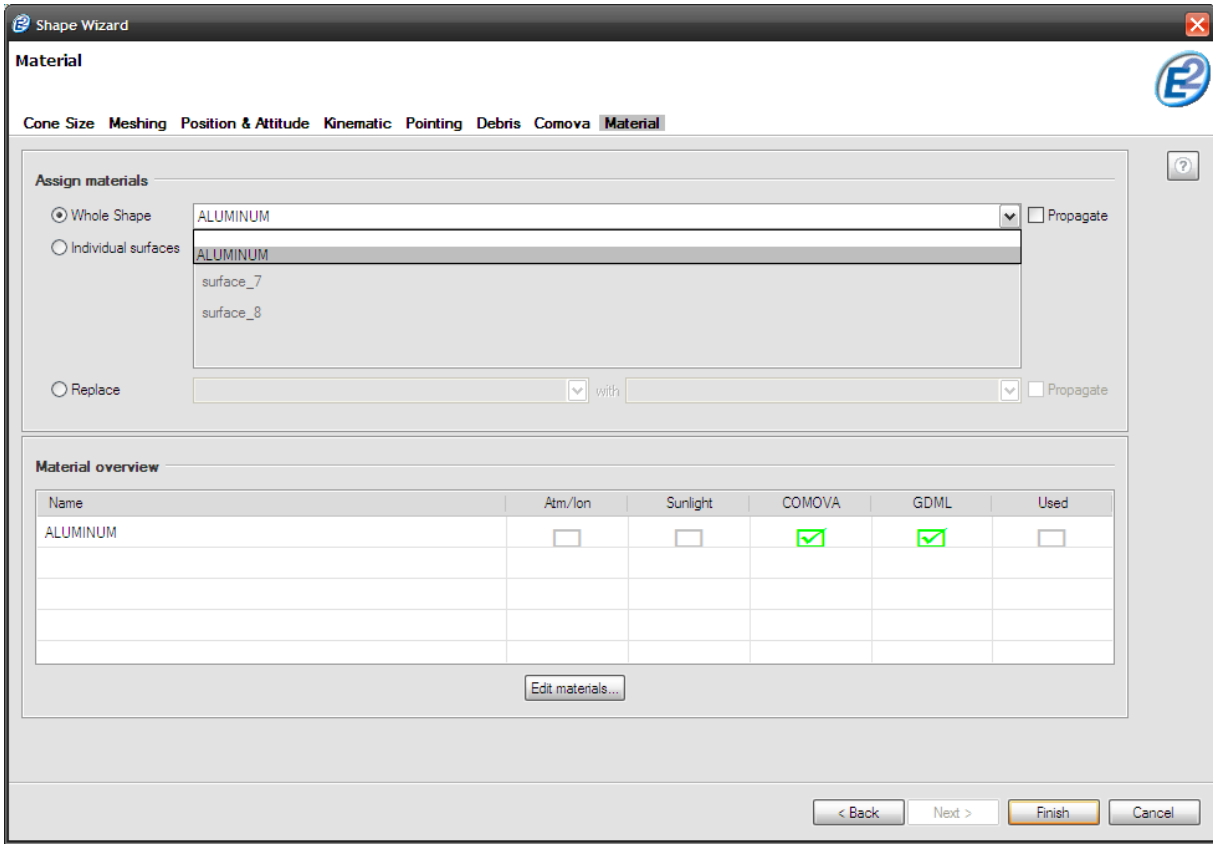

Figure 3-27: Material overview page, select a material for all surfaces

3.2.3.5 Antenna dish: Global Kinematics and Pointing

The following subsections are examples, but not full specifications of kinematics and pointing. For a full description of both functionalities, please see sections 3.3 and 3.4.

To extend our example satellite, we will add a parabole ("antenna dish"), which shall always point to earth. We start with the antenna dish (instead of the solar panels) because it will force the satellite system to rotate, and system pointing must always be defined before secondary (local) pointing.

Select the main body box in the 3D view and open the context menu to choose first "Coordinate Systems → global and local coordinate systems". You will now see the local coordinate system for each shape; this will be useful below.

Again, select the main body box and choose "Add Child → Parabole" from the context menu. Set p1 = (0, 0, 0), p2 = (0, 0, 20), diam = 80 mm and thickness = 2 mm. Then translate it by (50, 50, 0) and rotate it by (180, 0, 0). The result is shown in the following figure.

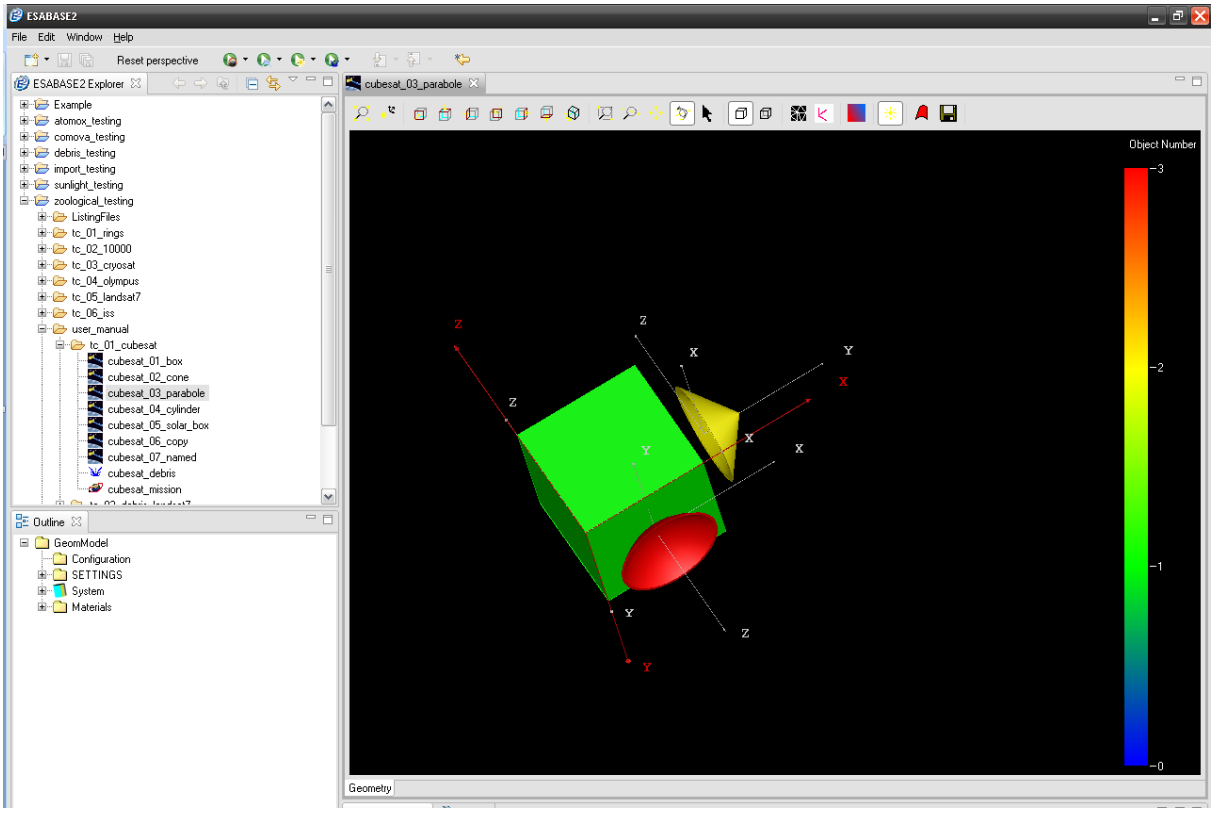

Figure 3-28: Geometry editor, example satellite with antenna dish at bottom

On the figure, the z axis of the parabole points "down", the x axis points into velocity direction and the y axis to the right side of the satellite. With this in mind, select the parabole and choose "Modify → Pointing" from the context menu.

The following figure shows the Pointing page for the parabole.

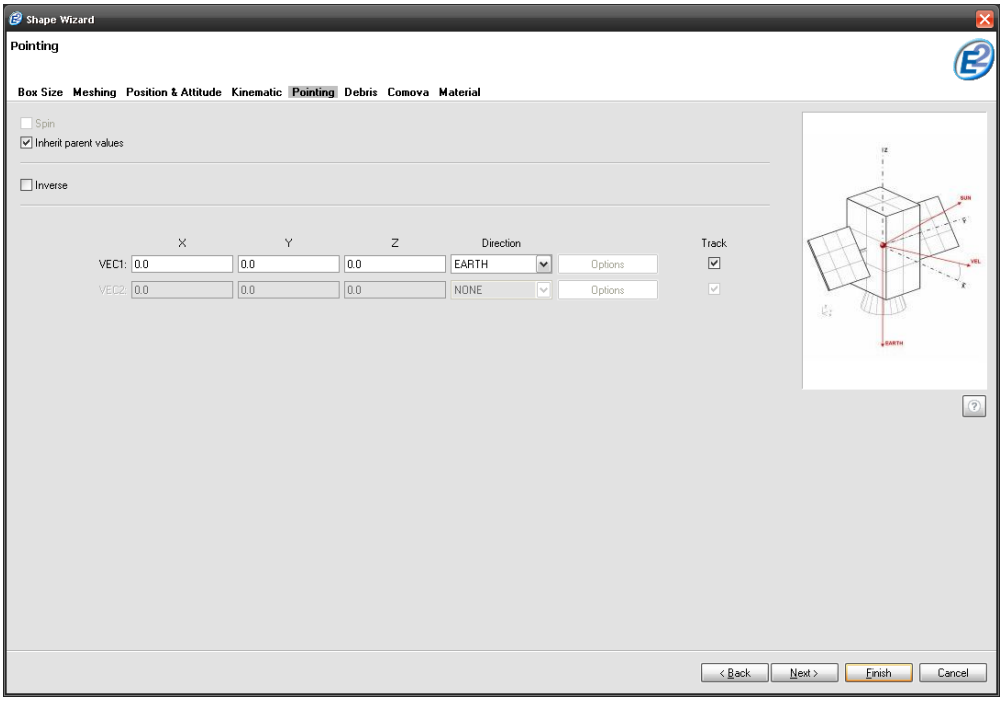

Figure 3-29: Geometry editor, Pointing page (antenna dish)

Set VEC = (0, 0, 1) and direction = EARTH. Also, activate the "Inherit from parent" checkbox; this will allow the parabole to force the parent body to rotate, if possible (more on that in a minute).

In ESABASE2, all shapes possessing pointing must also have kinematics. Although we do not wish to move the parabole (the satellite should move instead), it has to have at least minimal kinematic freedoms.

Kinematic operates on the concept of an axis around which to move (specified by a vector). Perpendicular to this axis, a second vector indicates where the 0° position is (i.e. the "natural" rotation angle). Then, a minimum and maximum angle is specified.

Up to 3 kinematic axes can be defined, although in our example, we will use only one.

The Kinematic page of the parabole wizard is shown in the following figure.

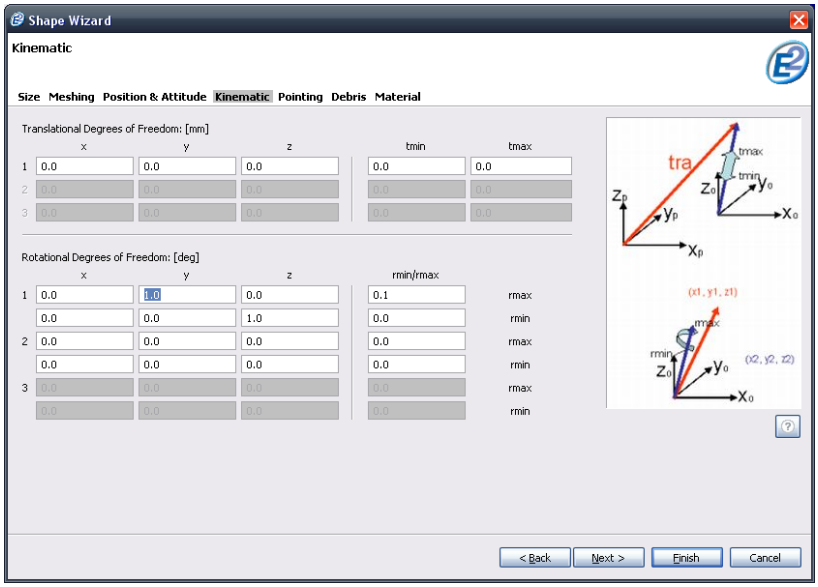

Figure 3-30: Geometry editor, Kinematic page (antenna dish)

We ignore the top section (translational kinematics; see section 3.3 for more) and go to the bottom section (rotational kinematics), which has three 2-line specifications: The first line for the axis vector, the second line for the 0° vector, and to the right the maximum and minimum angle (rmax/rmin).

Since having a pointing requires us to have at least minimal kinematics, set the first line to (0, 1, 0), the second line to (0, 0, 1), and rmax = 0.1 and rmin = 0. Then press "Finish".

The pointing and kinematics on the parabole are now configured to allow forcing the parent body to move. However, the parent body has to accept this: Therefore, we will edit the global system pointing.

Go to the Outline. You will see a node "System" here; rightclick it and choose "Modify Object". A wizard will appear – and unlike the standard geometry wizard, the System node wizard shows only pages for "Size", "Position and Attitude", and "Pointing".

Go to the Pointing page, which looks like portrayed by the following figure.

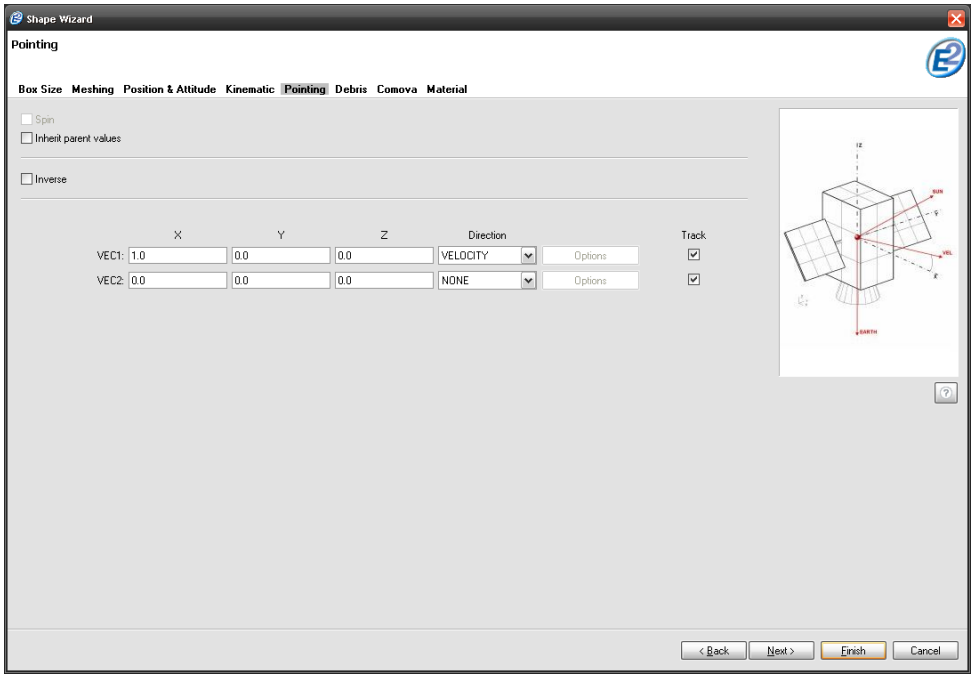

Figure 3-31: Geometry editor, Pointing page (System node)

On this page, velocity ("forward") and earth ("down") vectors are set by default. Since the parabole takes over the earth pointing, deactivate it here by setting VECB = (0, 0, 0) and direction = NONE. If you were not to do so, the parabole and the system pointing would come into conflict.

Click "Finish", and the global pointing is adjusted, complementing the local pointing performed on the parabole.

On a final note, please remember that system pointing must always come before secondary (local) pointing! Otherwise, the locally pointed shapes will be rotated before the system, leading to strange geometries at the orbital points in a solver result file.

3.2.3.6 Solar panel: Local Kinematics & Pointing

In the previous subsection, we performed global pointing for the antenna dish. Now we will concentrate on purely local pointing, i.e. the satellite orientation is fixed when seen from the local shape. A satellite panel is a good candidate for local pointing.

Select the main body box in the 3D view and open the context menu to choose "Add child → Cylinder". On the size page, set p1 = (0, 0, 0), p2 = (0, 30, 0) and diam = 20 mm. On the position and attitude page, set translation = (50, 100, 50). This will centre the cylinder on the left side of the satellite (when looking from the back).

This cylinder is the moving part within the solar panel, so we specify Kinematics and Pointing on this shape. The following figure shows the Kinematics page in the wizard.

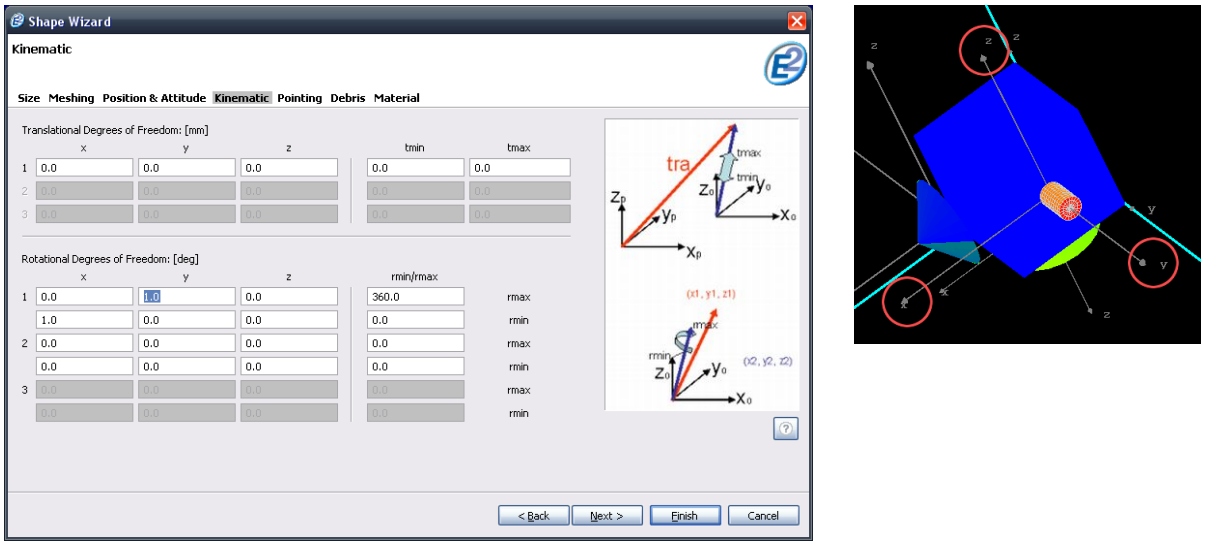

Figure 3-32: Geometry editor, Kinematic page (solar panel)

As we can see in the local coordinate system of the cylinder, we want to rotate around the y axis. Therefore, in the first line of the rotation section of the Kinematics page, set the rotation vector to (0, 1, 0), rmax = 360° and rmin = 0°.

In the second line, to define where exactly 0° is, choose (1, 0, 0), which is the vector along the x axis, pointing into the velocity direction.

Having specified the kinematics, we go to the Pointing page to specify the vector that should point to the sun, using the freedoms afforded by the kinematics. The Pointing page is illustrated in the following figure.

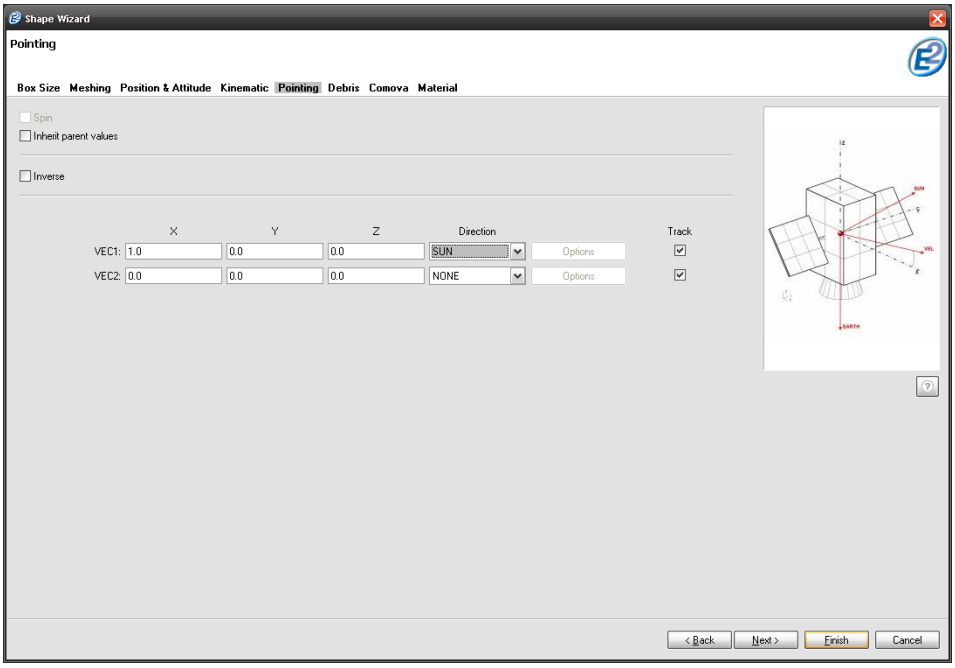

Figure 3-33: Geometry editor, Pointing page (solar panel)

Following the 0° kinematic vector specified above, choose the x-axis to point to the sun; set VEC1 = (1, 0, 0) and direction = SUN. Then "Finish" the wizard.

To complete the solar panel, select the cylinder and choose "Add Child → Box" from the context menu. Set the size to (20, 100, 400) and translate it by (-10, 30, -200). The latter will move the box to the end of the cylinder in y direction and centre it x and z wise.

Please note that the front flat side of the box points along the x axis of the cylinder, and therefore matches our pointing direction.

For better result accuracy, go to the Meshing page of the solar panel box. There, set number of elements for b-axis = 4 and c-axis = 16. Select the solar panel box again in the 3D view to see the new meshing.

Now repeat the process on the right side of the satellite. You might consider using the Copy & Paste functions in the Outline, explained in subsection 3.2.4. The result should be similar to the figure below.

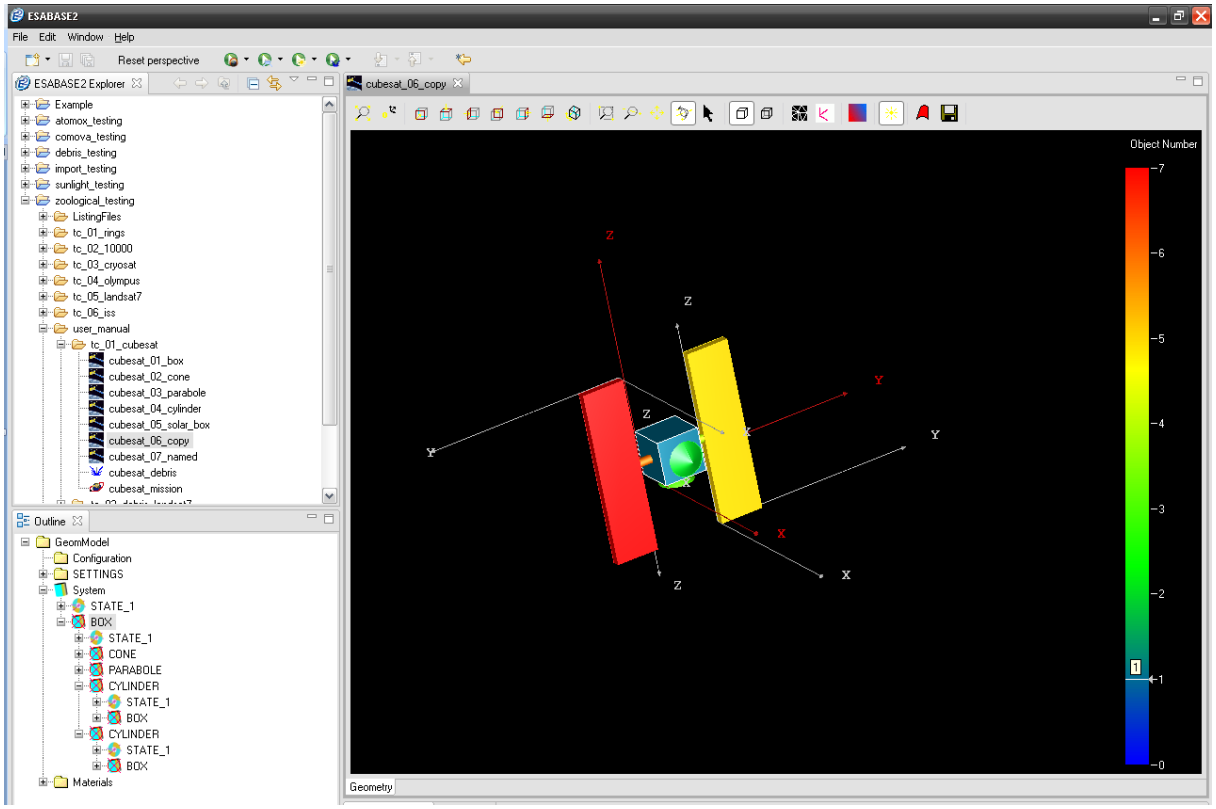

Figure 3-34: Geometry editor, example satellite with two solar panels

Please note that in the geometry editor, neither kinematics nor pointing is visualised in the 3D view. Only in the different solvers' result editors you can see the different configurations per orbital point. As an example, the following figure shows the orbital points of an ESABASE2/Debris run, using MASTER 2001 on a sun synchronous orbit.

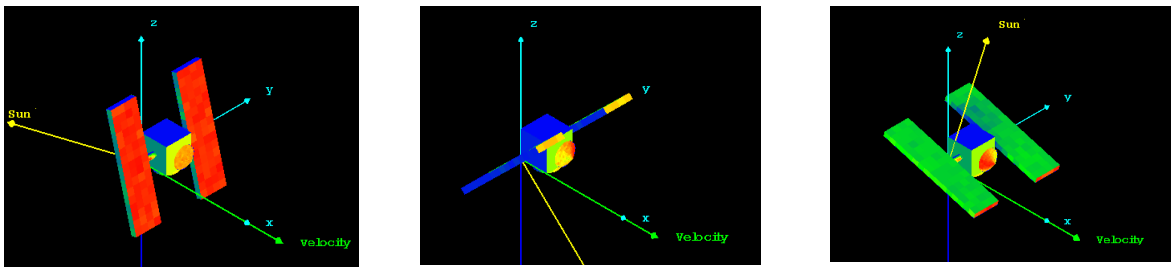

Figure 3-35: Orbital point configurations, example satellite

Read on to the next section to learn how to give shapes more intuitive names.

3.2.4 Geometry Outline and Properties View

In the Outline view, you can see the tree structure of the satellite geometry created in the previous subsections. This Outline is good for several things:

- *Shape Naming*: Allows you to rename shapes such as "BOX" to "Central body" or "Solar panel (left)".
- *Copy & Paste*: How to duplicate parts of the geometry model.
- *Shape Information*: See the exact contents of each shape, down to surface and meshed element level.
- *Shape Editing*: How to edit geometry values in the Property editor with the help of the Outline.
- *Enclosures*: A typical case for editing values in the property editor. Enclosures are advanced functionality that is not integrated into the Geometry editor.

3.2.4.1 Shape Naming

In the last subsection, you have defined a spacecraft geometry, represented as a 3D model. The shapes are ordered hierarchically, depending on the shape you performed "Add Child" upon. To see this hierarchy, consider the outline on the left side of the following figure.

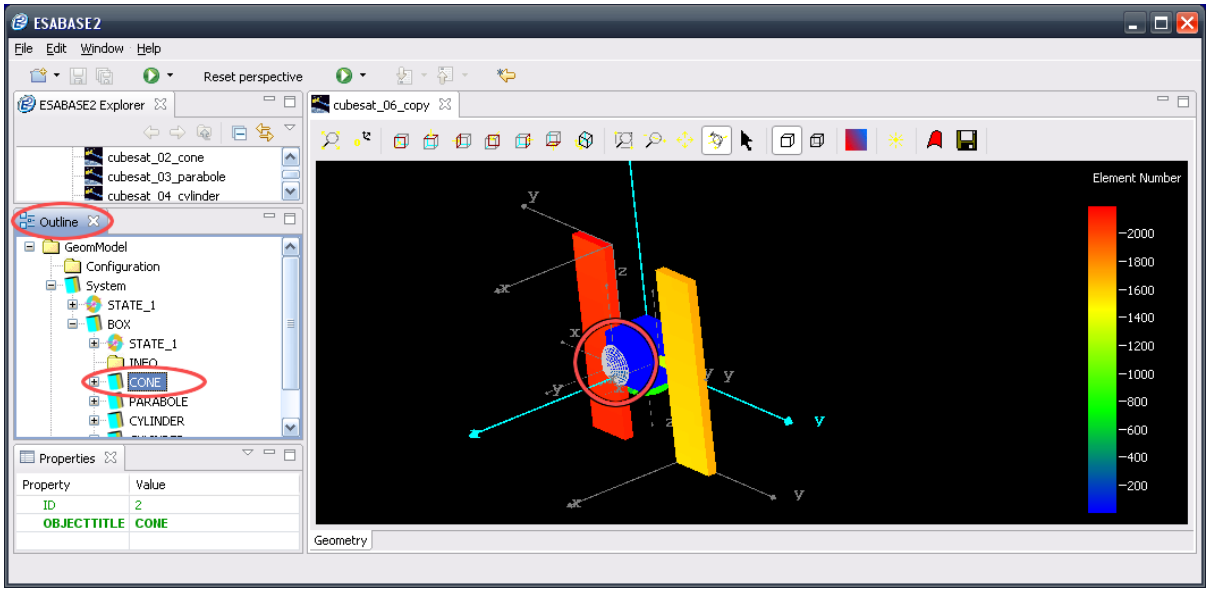

Figure 3-36: Geometry editor, Outline, Cone selected

From the "GeomModel" root node, you see the "System" node (where you specified the system pointing in chapter 3.2.3.5), and further below a hierarchy of BOX, then CONE, PARABOLE, CYLINDER and further shapes.

When you select "CONE" with a leftclick, the cone in the 3D area is highlighted, so you know which node belongs to which shape. However, a name like "CONE" is not really intuitive – to change that, rightclick in the node in the Outline and choose "Rename".

The following figure shows the resulting Rename dialog, where you can edit e.g. "Front shield" as a more appropriate name.

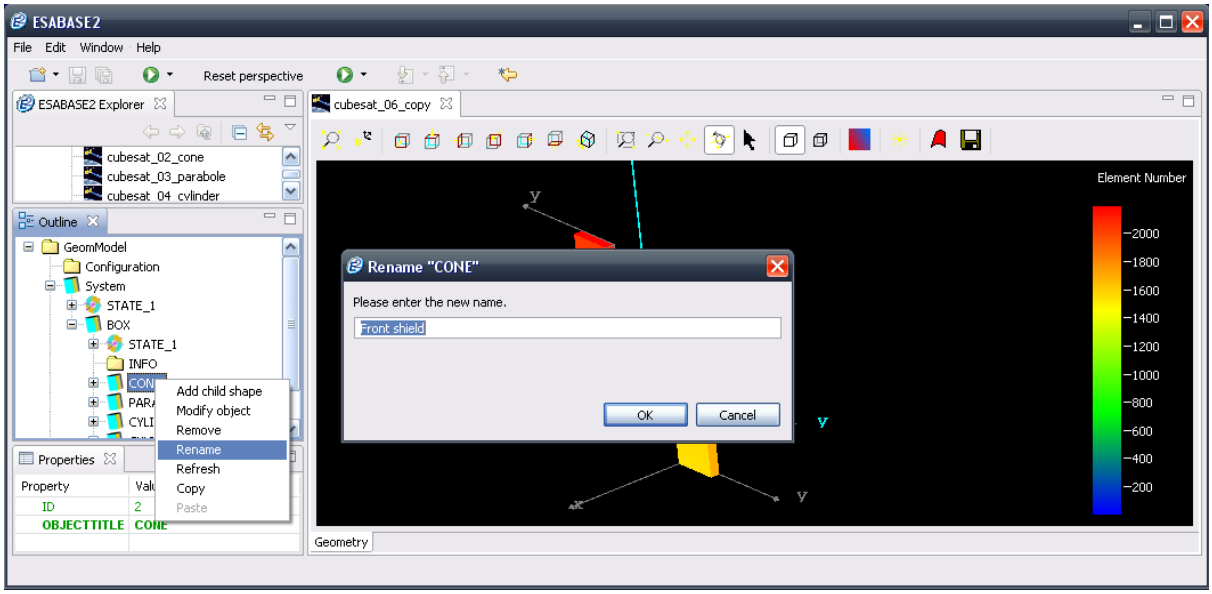

Figure 3-37: Geometry editor, Outline, Cone renaming

Continue in this manner and rename the "BOX" to "Central body", the "PARABOLE" to "Earth antenna", the cylinders to "Solar panel (left)" and "Solar panel (right)", and the cylinders' children "BOX" to "Panel".

Besides "Rename", other operations are available:

- "Add child shape" is the same as choosing "Add child → Shape" in the 3D view, when rightclicking a shape to invoke its context menu.
- "Remove" is the same as the "Remove" in the 3D view context menu of a shape.
- "Modify object" is the same as "Modify → Size" in the 3D view context menu of a shape, leading to the first page of the shape wizard.
- "Refresh" refreshes the Outline structure. In rare cases (usually involving the import of non-standard geometry files), this might be come necessary after modifying a shape.
- "Copy" & Paste is the subject of the next subsubsection.

The figure below shows the Outline with renamed shape nodes (left), and the context menu used to invoke the commands described above (right).

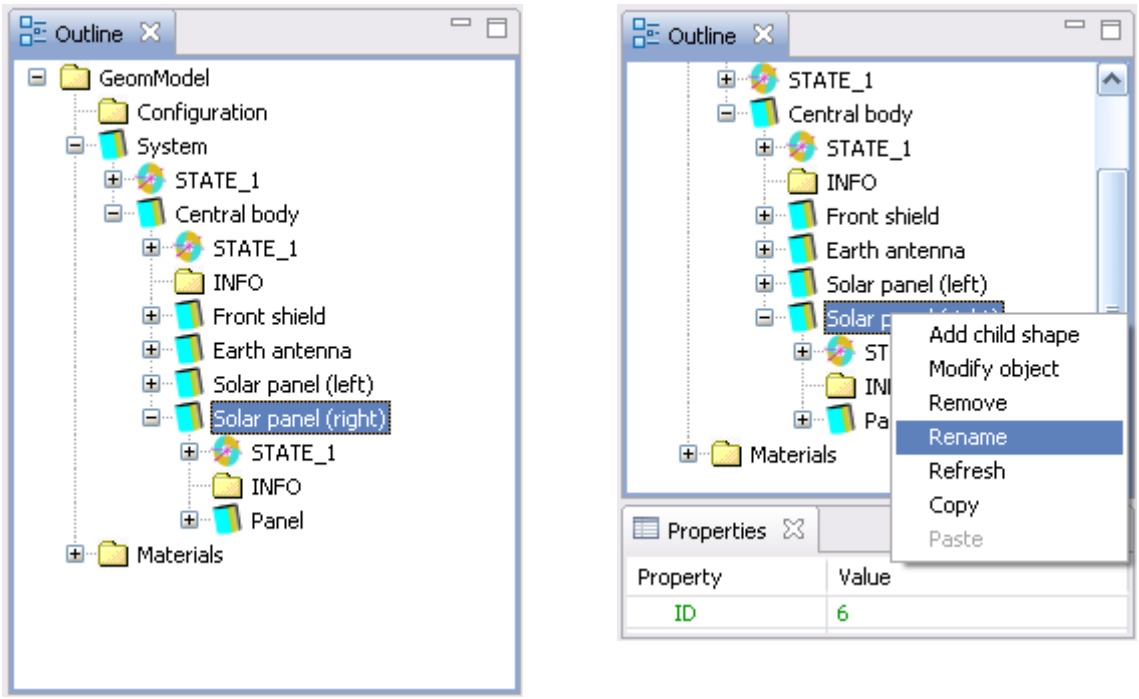

Figure 3-38: Geometry editor, Outline, node renaming

With the shapes now properly named, 3D view and Outline together represent a spacecraft geometry more intuitively than the 3D model alone can do.

3.2.4.2 Copy & Paste

The major reason to become familiar with the Outline is the Copy & Paste function. It is performed on the tree (instead of the 3D model) because positioning of shapes within the tree hierarchy can become a major issue if not properly observed.

To copy a shape for later pasting, go the Outline and rightclick a shape name (e.g. "BOX" or "Solar Panel") to open the context menu. Then, choose copy.

Both copy and paste actions are shown in the following figure.

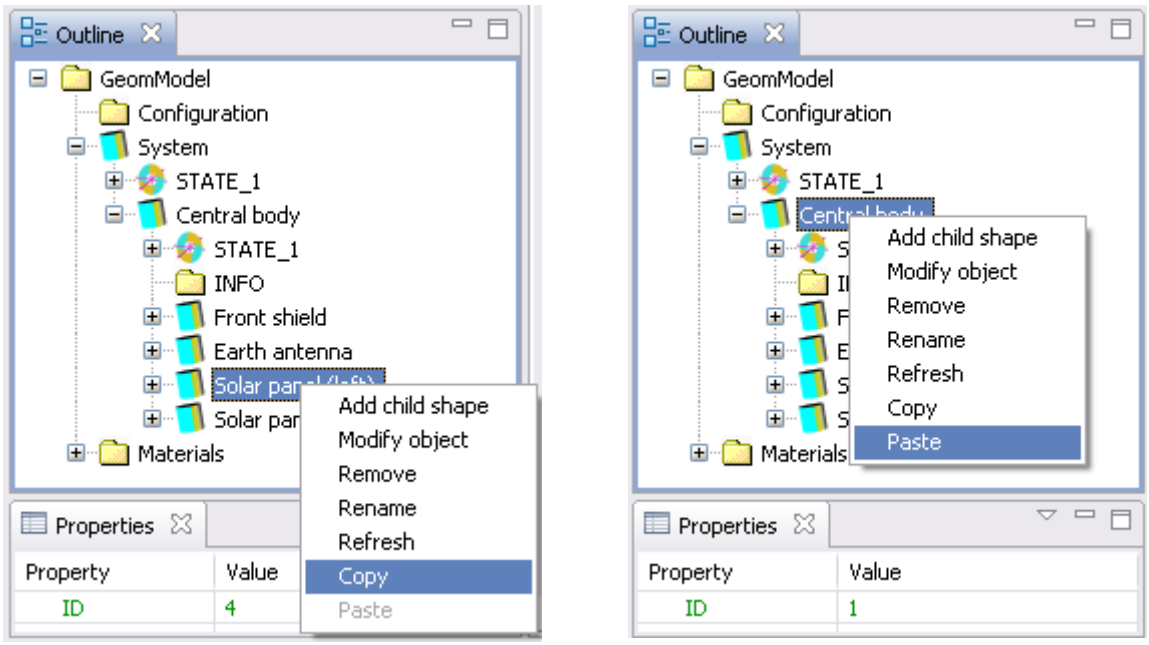

Figure 3-39: Geometry editor, Outline, Copy & Paste

On the left screenshot, the "Paste" menu entry is still disabled, because no shape has been copied into the buffer ("clipboard") yet. The shape is directly selected and copied.

On the right screenshot, the context menu has been opened on the parent element (the "Central body"). This is because the pasted object will be inserted as a child shape (analogous to "Add child shape"). If you were to perform the paste directly on the copied node, the pasted solar panel would appear as a child of the original solar panel, not as a sibling.

After a paste operation, the 3D view will not show any differences; the reason is that the pasted shape overlays the original one. Rightclick the new shape and choose "Modify object" from the context menu. Then, go to the "Position and Attitude" page to translate and rotate the new shape (see 3.2.3.3).

3.2.4.3 Shape Information

Our view of the Outline has been quite shallow yet. Expanding the tree nodes and at the same time looking at the Properties view can show you much about the internal data structures of a geometry file, as shown in the following figure.

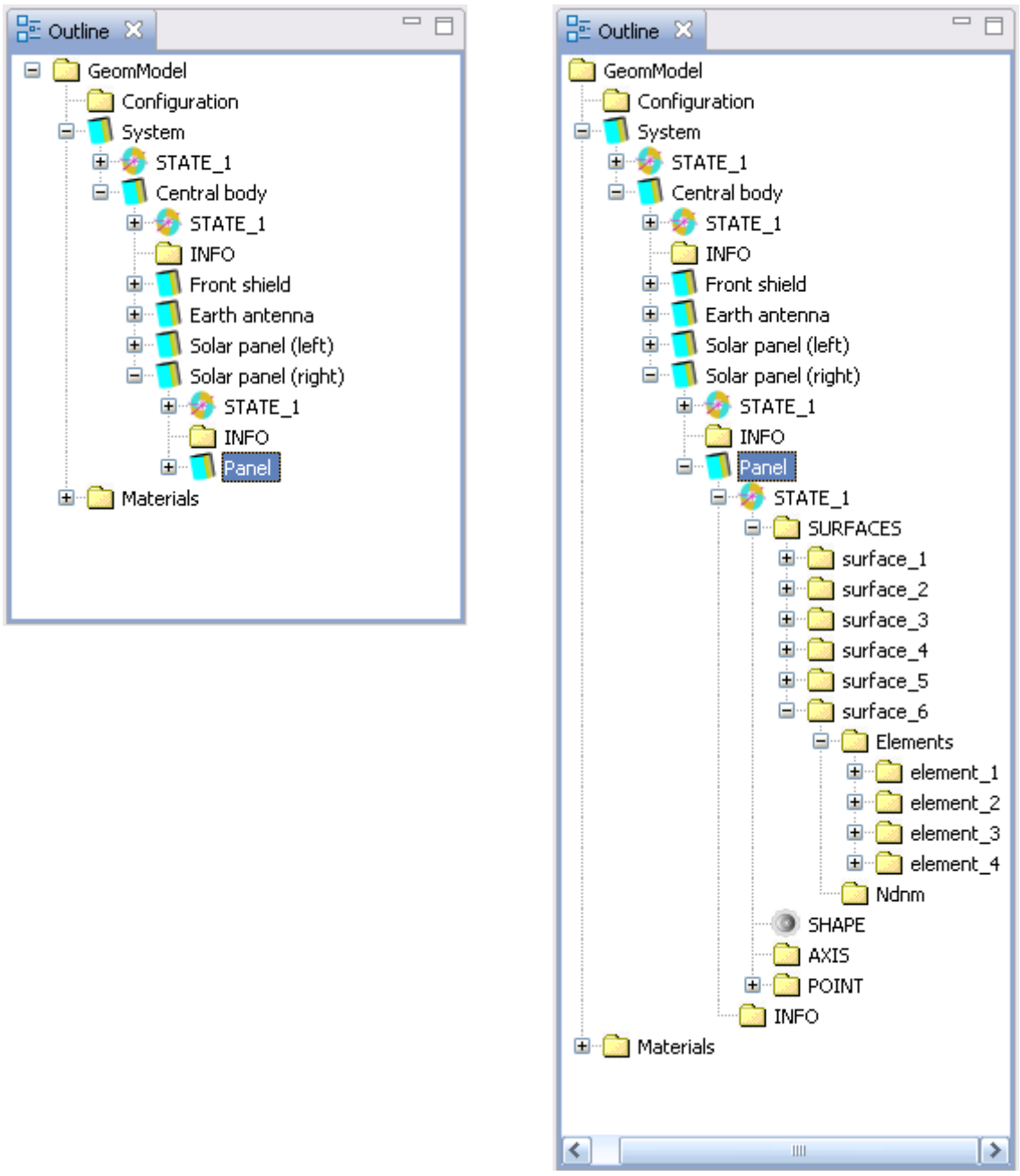

Figure 3-40: Geometry editor, Outline, expanded

The tree on the right is an expanded version of the tree on the left.

As the panel is a flat box, you can see its 6 surfaces; and below the last surface, its meshed elements (4 elements per surface). These are the elements whose coordinates will be used in Debris and other analyses.

When looking at the Properties view at the same time, you can see the data values for each node in the Outline, e.g. the element area and the normal vector for a meshed element, as shown on the left in the following figure.

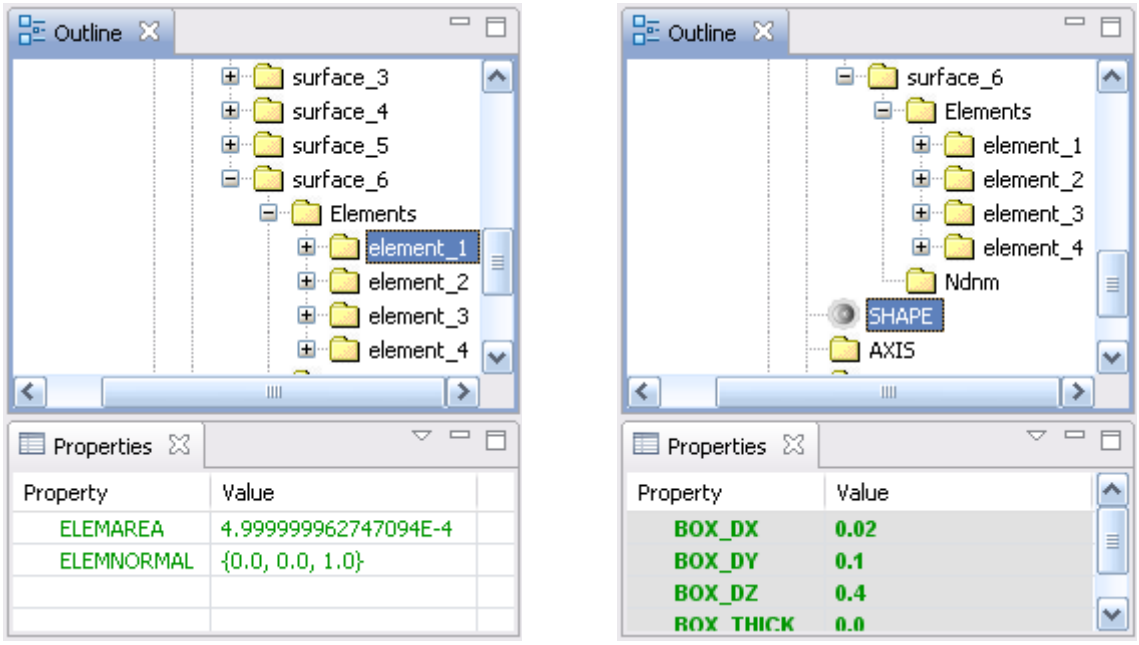

Figure 3-41: Geometry editor, Outline and Properties view

Other nodes are accessible in the same way. The right side of the figure above shows the "SHAPE" node in the Outline; the corresponding properties view shows for the panel: the x, y, z size and the box thickness.

The attributes you see here will appear familiar: These are the ones that you already edited in the Shape Wizard on the size page (see 3.2.3.1).

Please note that the amount of nodes and attributes you see in the Outline and Properties view depend on the user level setting made in the preferences (in the menubar, choose "Window → Preferences, then adjust "user mode").

The default setting is "Normal", where not all information is displayed. With the "Expert" setting, you can see more nodes and attributes. Finally, the "Developer" setting (may not be available in some versions) shows everything, but enables you to crash the application in creative ways, if you change data values (see next subsection) wrongly.

3.2.4.4 Shape Editing

Outline and Properties view not only enable you to see that data within a spacecraft geometry, but also give you the opportunity to edit attributes directly. Leftclick on a value in the Properties view to activate editing, as shown in the following figure.

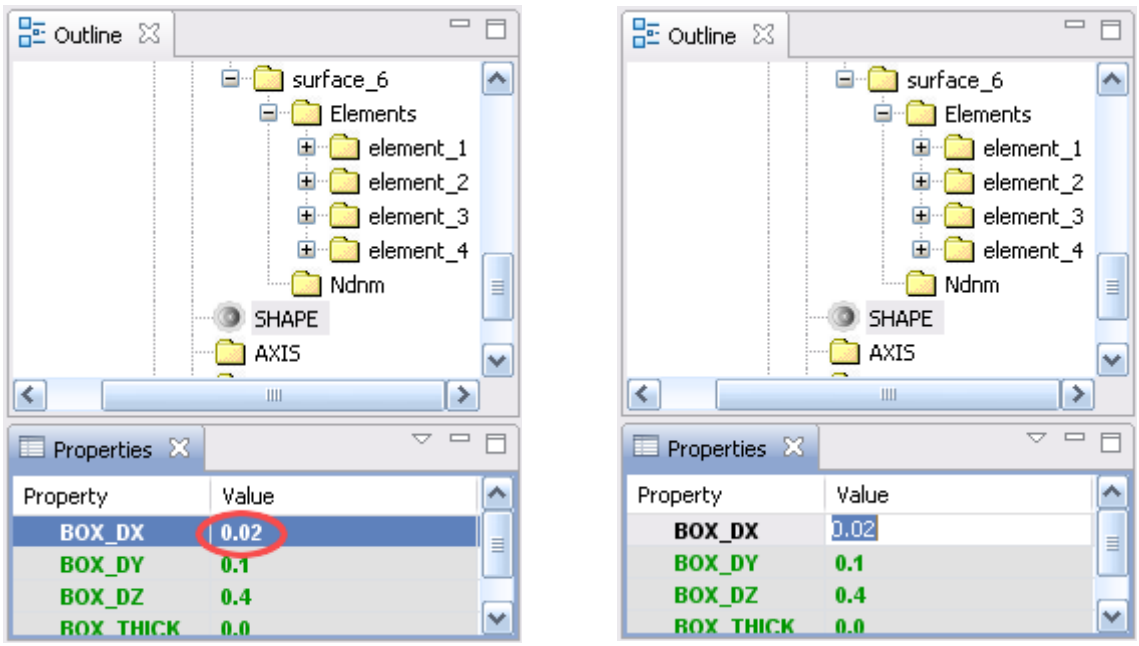

Figure 3-42: Geometry editor, Outline and Properties view, editing

On the left side, you see a red circle for the "BOX_DX" attribute. After the leftclick, the right side screenshot shows a text field for the value. You can enter any value here and press Return on the keyboard to make the change effective.

Please note that not all values may be editable; for some attributes (e.g. calculated ones), this has been turned off since it would never make sense to edit them.

3.2.4.5 Enclosures

Shapes can be grouped into so called "Enclosures". Each enclosure group has a numerical id (1, 2, 3 …), and some applications (e.g. Sunlight /5/) can operate on the enclosures.

You can define enclosures only by using the properties editor, as shown in the figure below (and described in the previous subsection (3.2.4.4)).

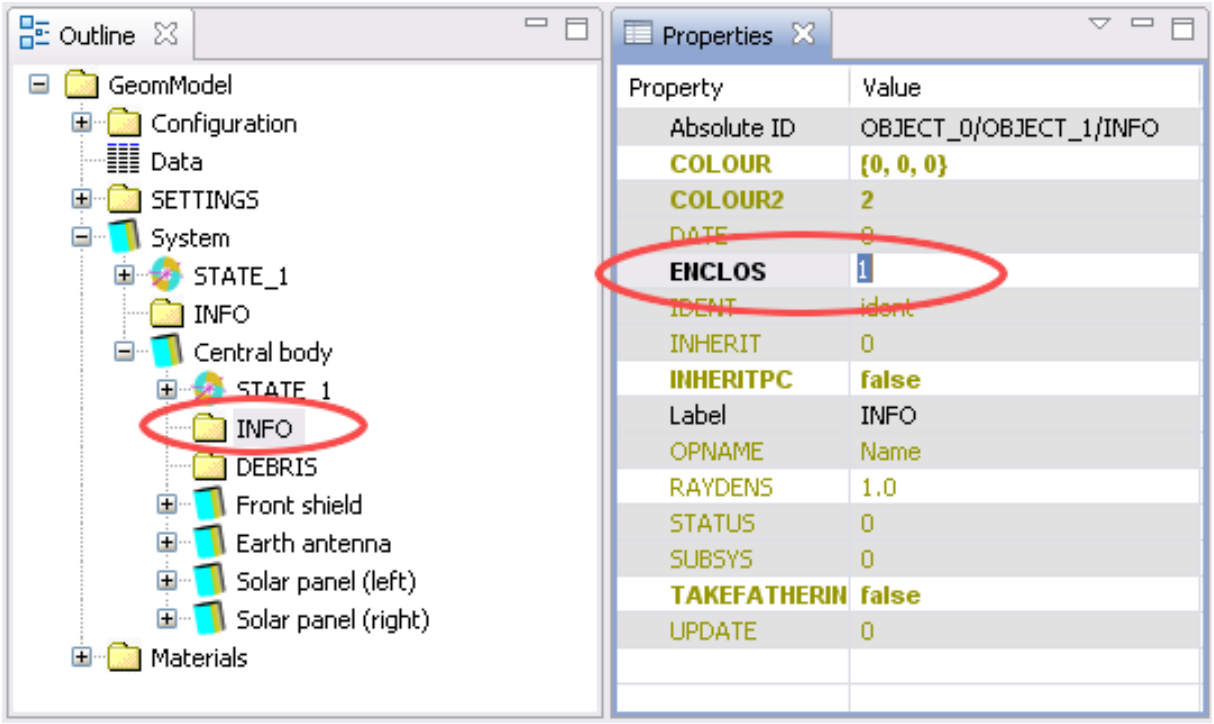

Figure 3-43: Geometry editor, Outline and Properties view, Enclosure definition

By default, all shapes belong to enclosure 1. To change the id, select the shape in the Outline, navigate to the "Info" node, and – in the Properties view – edit the value of "ENCLOS".

Applications that do not know what enclosures are will ignore this parameter.

Please note that the "Info" node (and the "ENCLOS" parameter) will only be visible to you in the Expert mode (to select Normal or Expert mode, in the menu bar, choose "Window → Preferences → ESABASE2").

3.2.5 Boolean Operations

The starting point of a Boolean Operation is two overlapping 3D objects /10/. The available Boolean Operations which can be performed are Intersection, Union and Subtraction.

The following graphic depicts a cone which has a sphere as child shape.

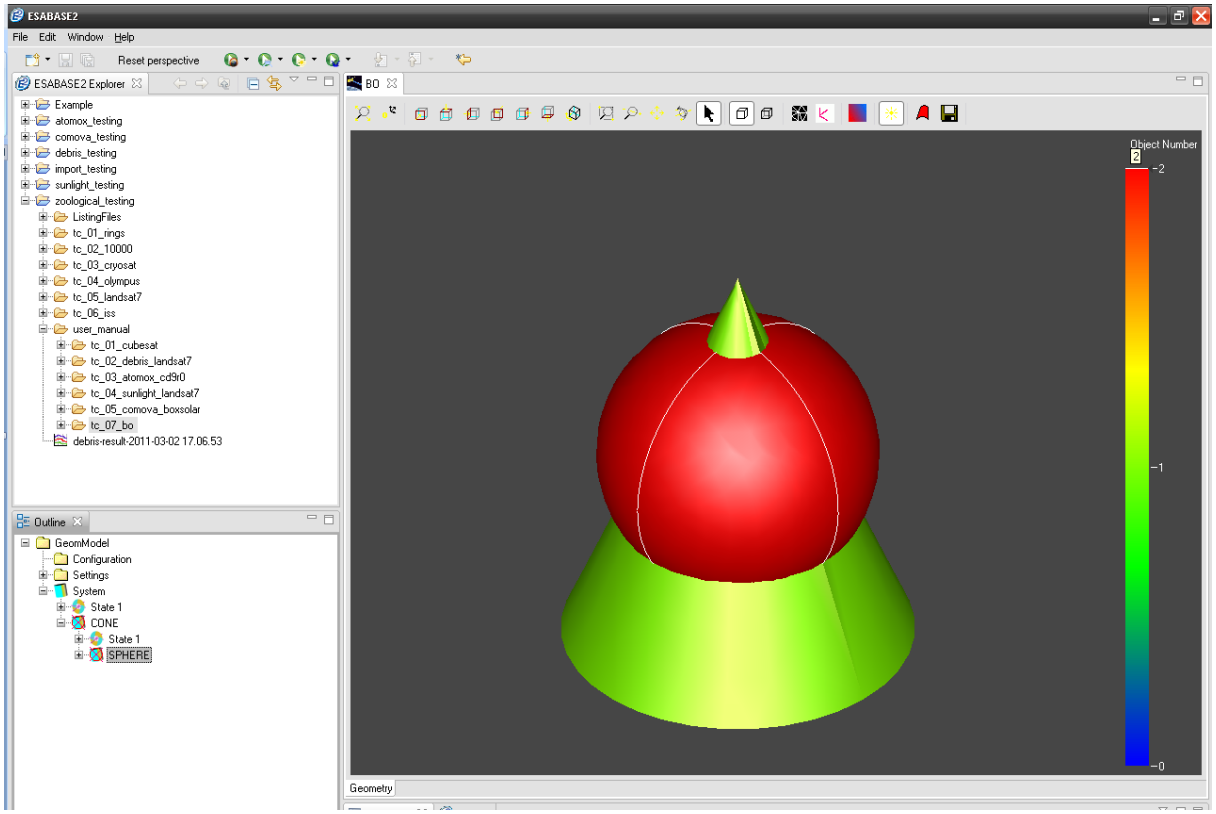

Figure 3-44: Geometry editor, Outline and Properties view, Enclosure definition

To perform a Boolean Operation, select the child shape in the outline and open the context menu. The menu will display the three different BO.

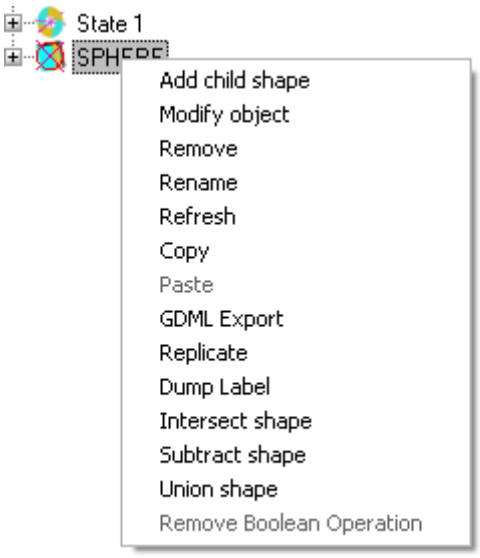

Figure 3-45: The context menu in the outline, containing the BO

Select "Intersect". The editor now displays the result of this operation:

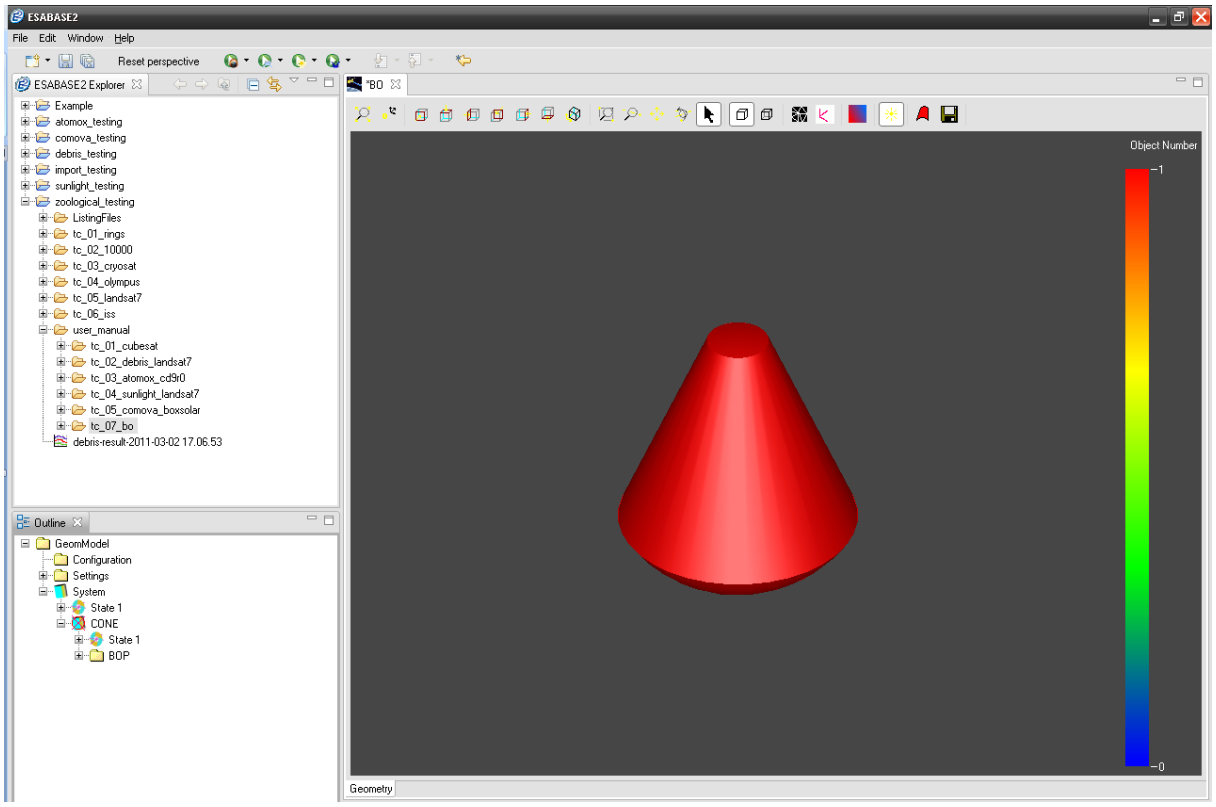

Figure 3-46: The shape after performing the operation

It is important to know that the resulting shape takes all secondary attributes like shielding or material settings from the parent node.

Let us assume that we are not satisfied with the result, and that we would like to decrease the radius of the sphere. This can be done via the outline. There, we find a node called "BOP" under the "CONE" object. Under the BOP node, we find a node called OBJECT_1_1. This is the sphere which was used as second operand in the BO.

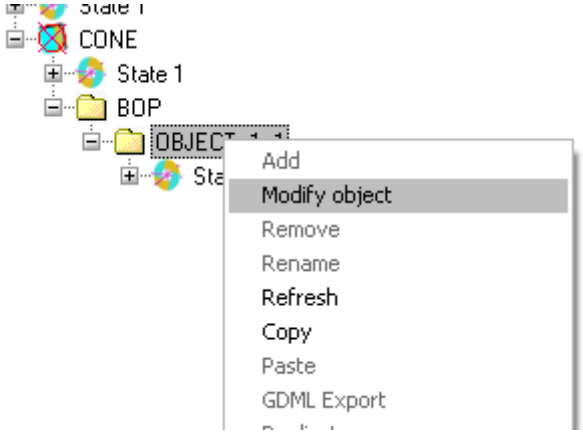

Figure 3-47: Modifying a BO via the outline

We select "Modify Object". The shape wizard opens. There, we decrease the radius of the sphere and press "Finish". The result is depicted next.

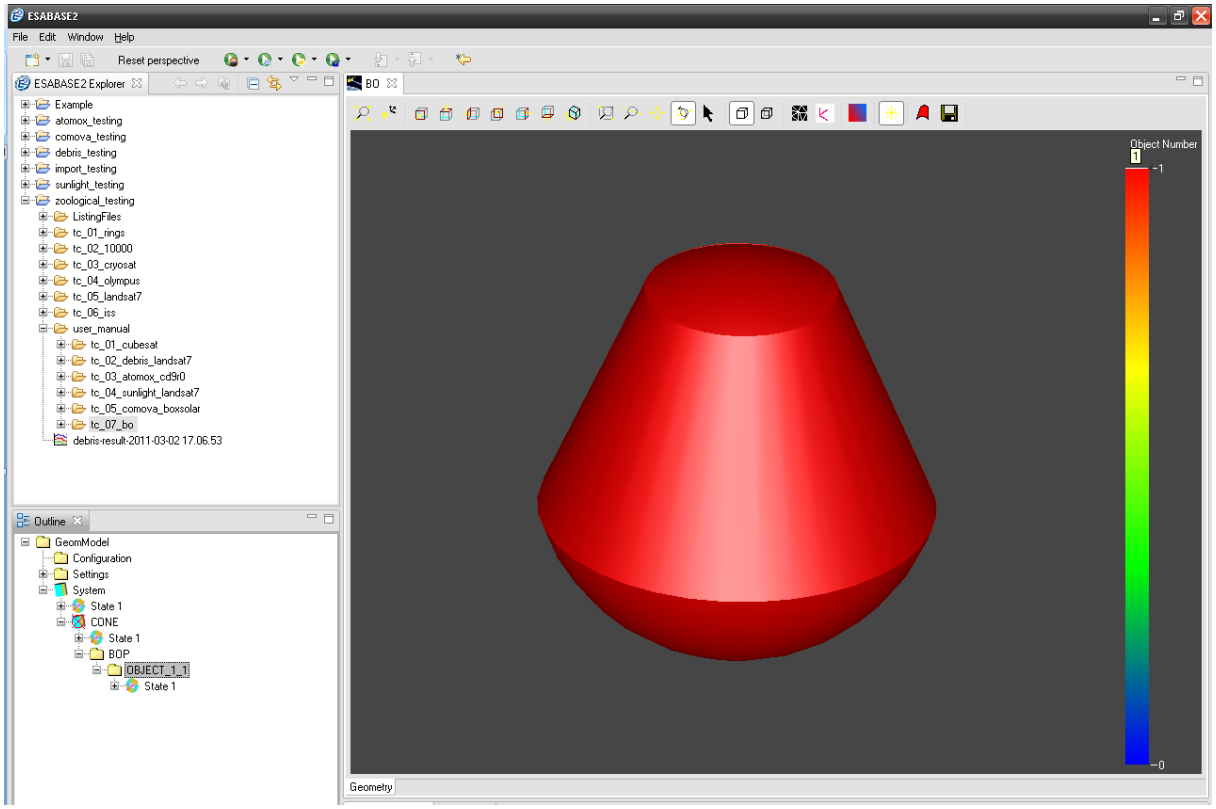

Figure 3-48: The shape after the modification

The other BO can be used in a similar way. BO can be chained, so it is possible to apply the next BO to this shape now by adding another child object and perform the operation in identical manner.

To undo a BO, one can select the second operand node of the BO under the "BOP" node, open the context menu and choose "Undo Boolean Operation". This restores the parent object, and the child object is deleted. If no more nodes are under the "BOP" node, this node is deleted also.

3.2.6 Geometry Model Import Interfaces

The ESABASE application offers four different possibilities to import geometries:

- *STEP File Import*: The STEP file format is an industry standard for exchanging data in files /7/. For ESABASE2, relevant formats are STEP AP-214 and AP-203.
- *BAS File Import*: Used by the original ESABASE, many spacecraft geometries defined in the BAS format can be re-used in ESABASE2.
- *GDML File Import*: GDML is a popular geometry model format in the radiation domain.
- *STEP-SPE File Import*: STEP-SPE is a protocol for geometry models developed by ESA in the course of several activities. It is a combination of STEP-TAS and STEP-NRF.

From STEP-TAS, the geometry definitions are included in STEP-SPE, and STEP-NRF contributes to this new protocol by providing means to define generic data structures.

3.2.6.1 STEP File Import

As of ESABASE2 version 4.0.0, the STEP AP-214 protocol is supported, but not AP-203 (STEP-SPE) yet. Also, please note that the size and form of imported STEP shapes cannot be changed within the geometry editor; the reason is that STEP shapes translate to an element mesh but not necessarily to attributes needed by the shape wizard.

To import a STEP AP geometry, create a new geometry file by using the menubar; choose "File → New → Geometry". The resulting wizard is shown in the following figure.

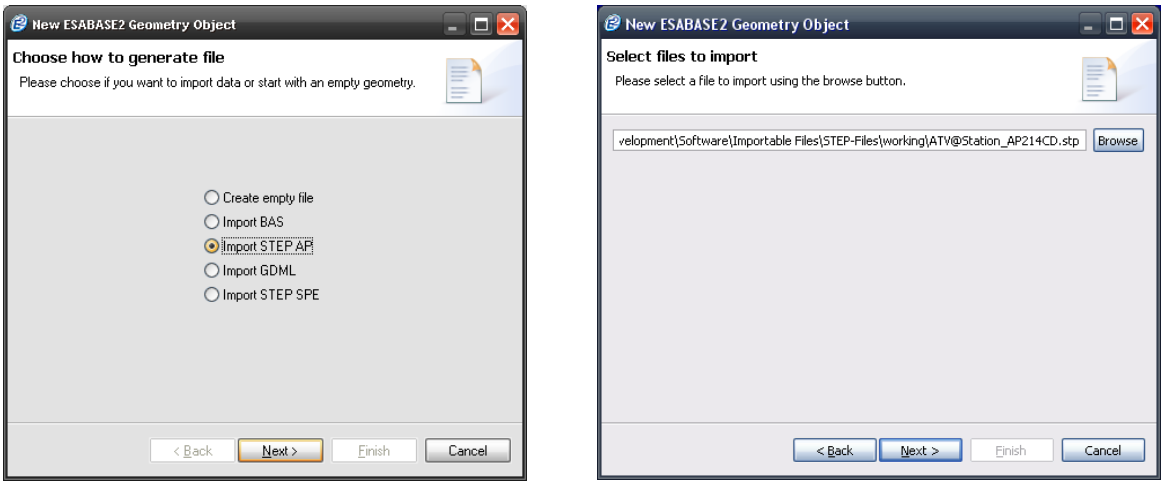

Figure 3-49: Geometry file creation wizard, STEP variant

On the first page, choose "Import STEP AP" in place of the standard "Create empty file". Then, on the next page, choose the input STEP file. The third page will ask you for a directory location, the same as for empty new geometries.

The following figure shows an ATV model imported from a STEP AP-214 file.

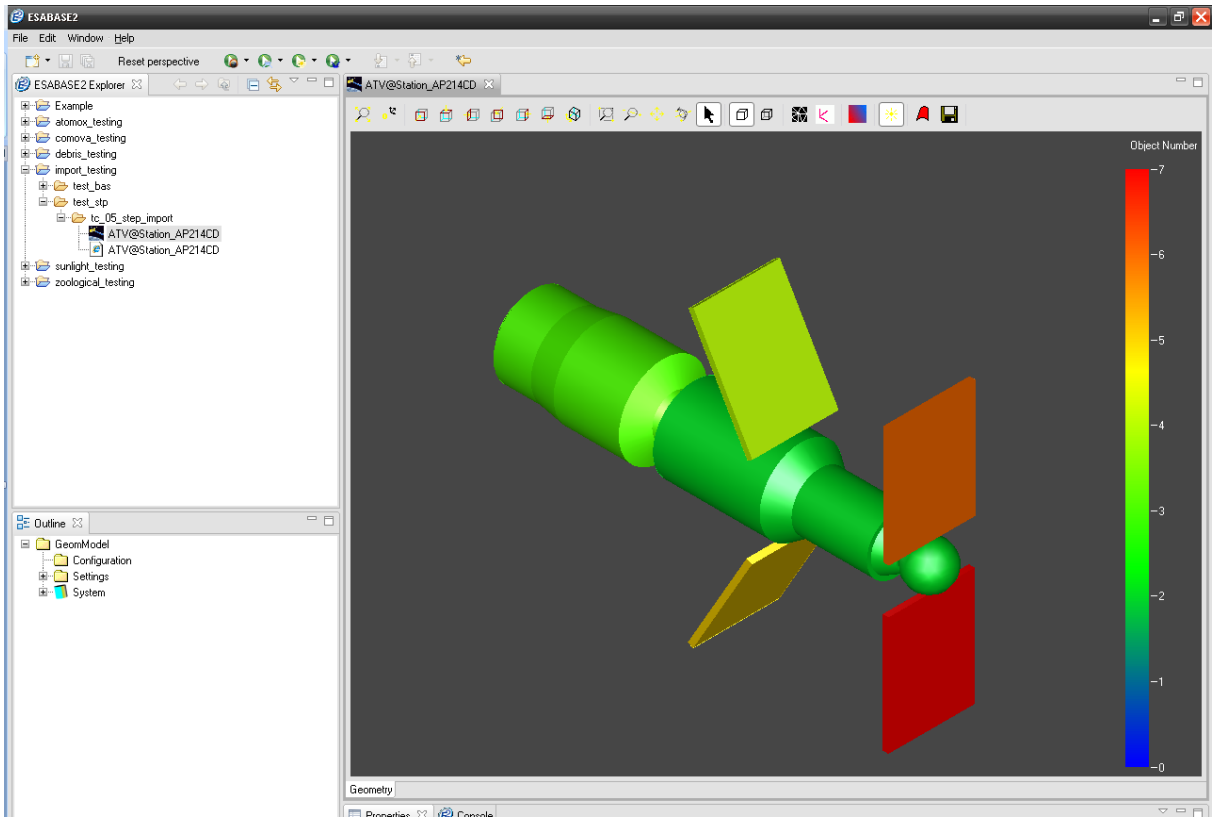

Figure 3-50: Geometry editor, imported STEP ATV model

The NetGen mesher is stored as default for every object. Occasionally, a STEP file is structured in a way that does not allow NetGen to automatically figure out a good way to mesh it. In this case, you can change the meshing parameters and/or the mesher via the Geometry Wizard for each object.

Finally, please note that only geometrical and topological STEP entities as well as assembly structures are imported into the geometry file. This means that you must re-specify the following information:

- Kinematics (see 3.2.3.5)
- Pointing (see 3.2.3.6)

Both can be done by using the ESABASE2 shape wizard.

3.2.6.2 BAS File Import

Users of the original ESABASE can re-use their geometry BAS files in ESABASE2 using the BAS import. In the menubar, choose "File → New → Geometry" and consider the wizard shown in the following figure.

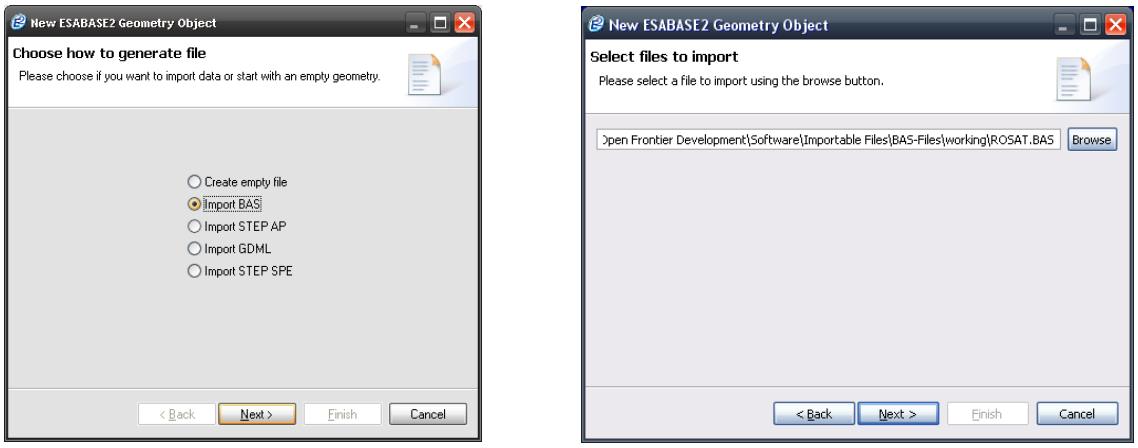

Figure 3-51: Geometry file creation wizard, BAS variant

As with STEP, only the pure geometry is imported (excluding kinematics and pointing, as this was specified in CMD files). Also, please note that certain constructs are not supported (see section 5.4.1); and that the file name of the BAS file may not exceed 12+3 characters.

BAS shapes can be edited normally in the Geometry editor; they are the same as shapes created with the shape wizard. Notable exceptions are facets which are not based on points (i.e. circle, line, polynom); only points are supported by the shape wizard in the geometry editor. Meshing, however, works on all imported shapes.

The figure below shows a model of the HUBBLE Space Telescope imported via BAS file.

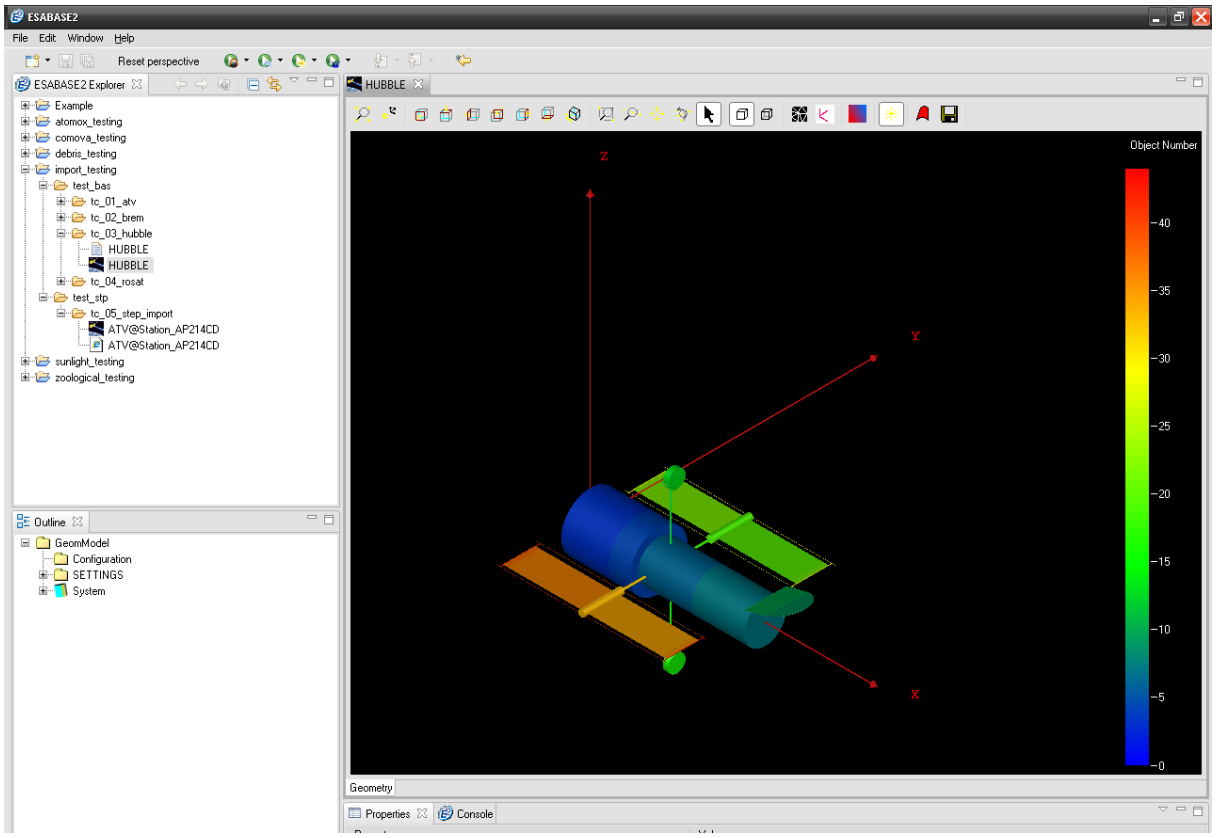

Figure 3-52: Geometry editor, imported HUBBLE.BAS

A sensible way to convert BAS files is to import them, check the existence of all shapes (COPY statements will be translated into duplicated objects in the ESABASE2 geometry file), and re-apply kinematics and pointing.

Please note that for some complex BAS geometries, some open issues exist (refer to section 5.4.1 for more details).

3.2.6.3 GDML File Import

As of ESABASE2 version 5.0.0, the GDML protocol is supported for import. As most of the shape from the GDML protocol can be mapped to the ESABASE2 shape palette, it is possible to edit the shapes dimension after the import.

To import a GDML geometry, create a new geometry file by using the menubar; choose "File → New → Geometry". The resulting wizard is shown in the following figure.

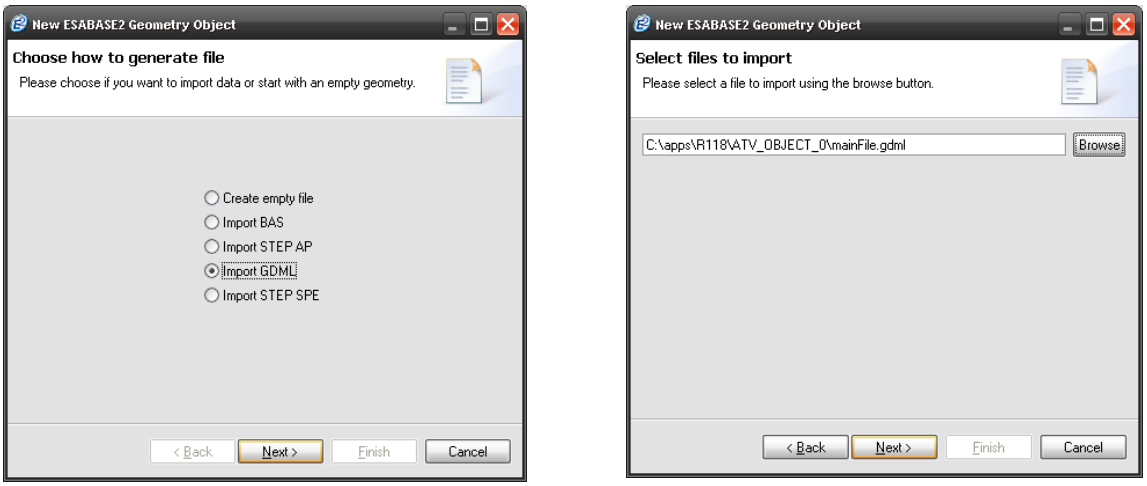

Figure 3-53: Geometry file creation wizard, GDML variant

On the first page, choose "Import GDML" in place of the standard "Create empty file". Then, on the next page, choose the input GDML file. The third page will ask you for a directory location, the same as for empty new geometries.

The NetGen mesher is stored as default for every object, but it is also possible to use the OCC mesher, or, if the shape is part of the ESABASE2 shape palette, also the ESABASE mesher.

Please note that for the GDML import an internet connection is required, due to the needed retrieval of the GDML schema from the website defined in the GDML file.

3.2.6.4 STEP-SPE File Import

As of ESABASE2 version 5.0.0, the STEP-SPE protocol is supported for import. As most of the shape from the STEP-SPE protocol can be mapped to the ESABASE2 shape palette, it is possible to edit the shapes dimension after the import.

To import a STEP-SPE geometry, create a new geometry file by using the menubar; choose "File → New → Geometry". The resulting wizard is shown in the following figure.

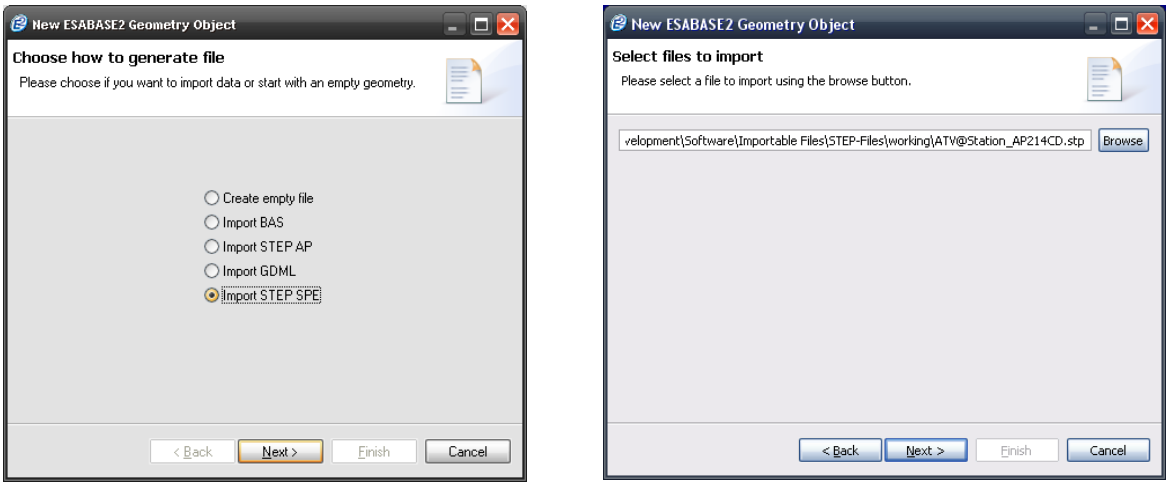

Figure 3-54: Geometry file creation wizard, STEP-SPE variant

On the first page, choose "Import STEP-SPE " in place of the standard "Create empty file". Then, on the next page, choose the input STEP-SPE file. The third page will ask you for a directory location, the same as for empty new geometries.

The NetGen mesher is stored as default for every object, but it is also possible to use the OCC mesher, or, if the shape is part of the ESABASE2 shape palette, also the ESABASE mesher.

3.2.7 Geometry Model Export Interfaces

3.2.7.1 GDML Export

As of ESABASE2 Release 4.0, this new functionality is supported by the application. GDML is a geometry data format, popular in the domain of radiation analysis (Geant4). To support the interoperability between the different tool frameworks, ESABASE2 offers you to export your geometry model into the GDML format. This section covers how this export functionality is used.

For this example, we have imported the ATV geometry from a BAS file. This BAS file is included in the ESABASE2 distribution. The following graphic depicts the spacecraft.

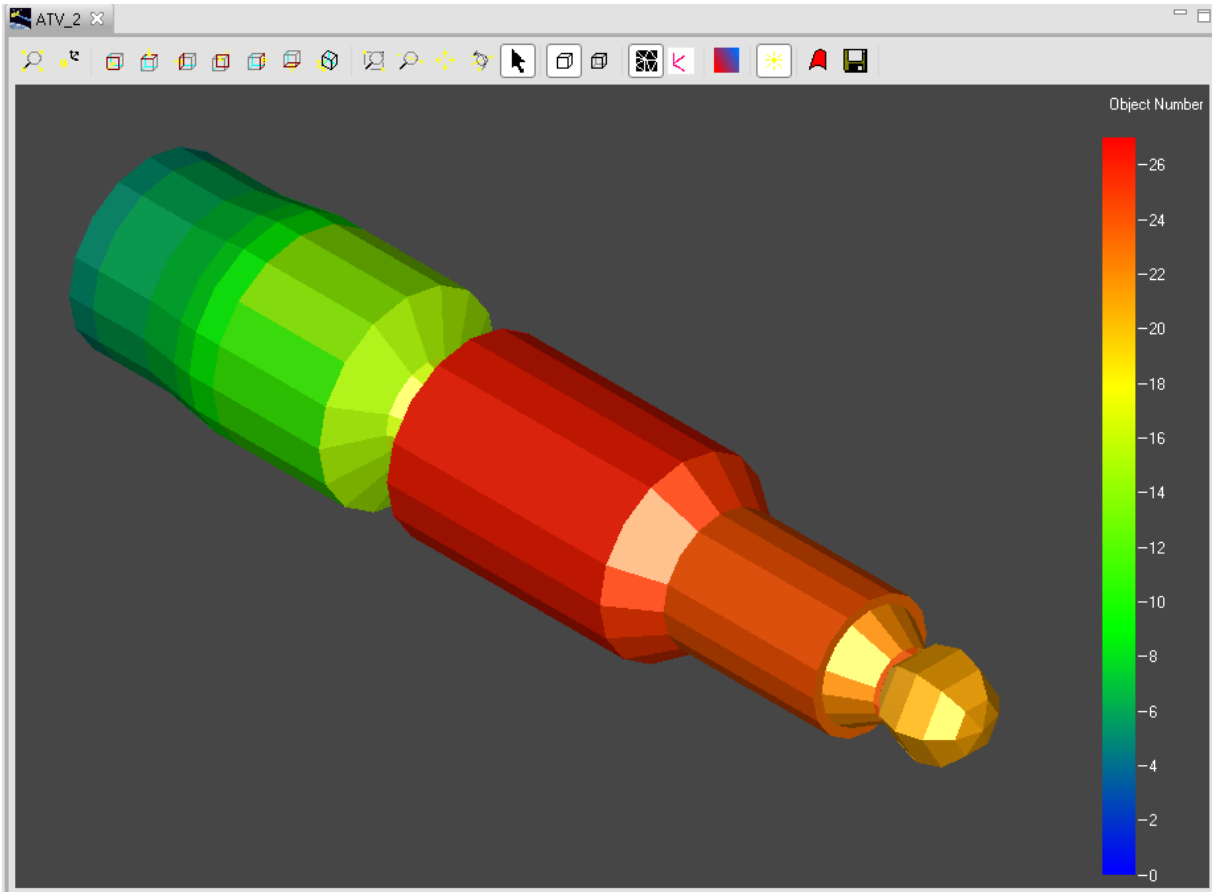

Figure 3-55: The ATV imported from a BAS file

To export this geometry to GDML format, two prerequisites have to be met:

1.) A material shall be assigned to each object to be exported. The GDML format requires a defined material for each exported object. If no material was assigned, a default material will be set in the resulting GDML file to produce a valid GDML file.

2.) The geometry shall be meshed. The GDML format supports only a subset of ESABASE2 shapes. If one of the shapes to be exported is not directly supported by GDML, the corresponding shape is exported as so called tessellated object. This means that the mesh elements are written to GDML instead of a geometry definition. This approach of course requires a present mesh.
If you try to export geometry objects that are not GDML compatible and not meshed manually, OCC mesher is performed meshing before exporting to GDML file. The section 5.4.3 describes which meshers are suitable for which shapes and why GDML exporter uses the OCC mesher as default mesher.

If the geometry is configured correctly, the export can be started. The export can be performed either for the whole system or for single objects (including their child objects). To start an export, select the respective object in the outline. In this example, the whole geometry shall be exported, so the system node is selected and the context menu is opened.

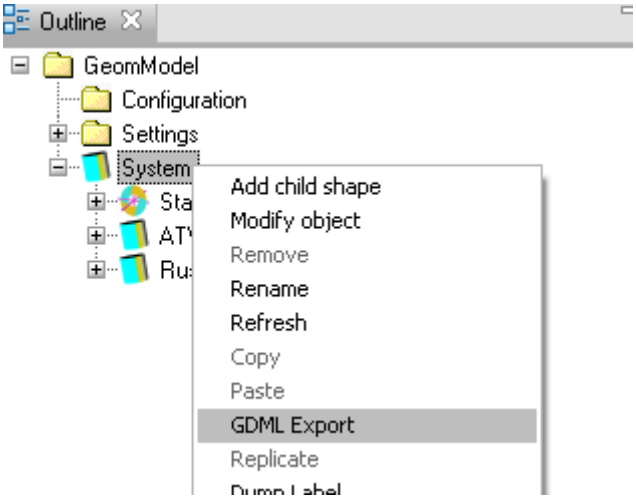

Figure 3-56: The GDML export entry in the context menu

Use the entry "GDML Export". A file chooser opens and asks for the directory into which the geometry shall be exported.

Figure 3-57: The directory for the export is selected

Please note that the exporter will create a subdirectory within the selected directory. The name of this directory is a combination of the file name and the object id. So for our example, a new directory is created within "GDML_Exports" with the name "ATV_OBJECT_0". Within this directory, a file named "mainFile.gdml" can be found, containing the structure description of the GDML data. Next to the main file, other data files are created on which the main file relies (Figure 3-58).

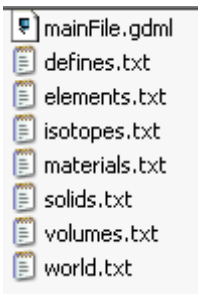

Figure 3-58: Files generated during GDML export

In detail, the main file is a GDML file that refers to other exported files from its several sections. This way e.g. the position information is stored separated from the shape information. This is done via XML entities. So when reading the GDML file with a standard-conform XML parser, the entities are included during reading. This way a program like Geant4 can operate on the full GDML data structure. The advantage of this split storage appears in case of large exports. It requires a smaller amount of memory during export and prevents from crashes due to out of memory errors.

The typical content of a "mainFile.gdml" is shown in Figure 3-59.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE gdml [
<!ENTITY defines SYSTEM "defines.txt">
<!ENTITY isotopes SYSTEM "isotopes.txt">
<!ENTITY elements SYSTEM "elements.txt">
<!ENTITY materials SYSTEM "materials.txt">
<!ENTITY solids SYSTEM "solids.txt">
<!ENTITY volumes SYSTEM "volumes.txt">
<!ENTITY world SYSTEM "world.txt">
]>
<gdml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
noNamespaceSchemaLocation="http://service-spi.web.cern.ch/service-spi/app/
releases/GDML/schema/gdml.xsd">
<define>
&defines;
</define>
<materials>
&isotopes;
&elements;
&materials;
</materials>
<solids>
&solids;
</solids>
<structure>
&volumes;
&world;
</structure>
<setup name="OF_setup" version="1.0">
<world ref="World"/>
</setup>
</gdml>
```

Figure 3-59: Typical content of "mainFile.gdml"

3.2.7.2 STEP-SPE Export

With release 5.0 of ESABASE2, it is possible to export geometry objects to files following the STEP-SPE protocol.

The export mechanism for STEP-SPE follows the same principles as the GDML export described in section 3.2.7.1. We use the simple satellite with two solar panels to demonstrate the export.

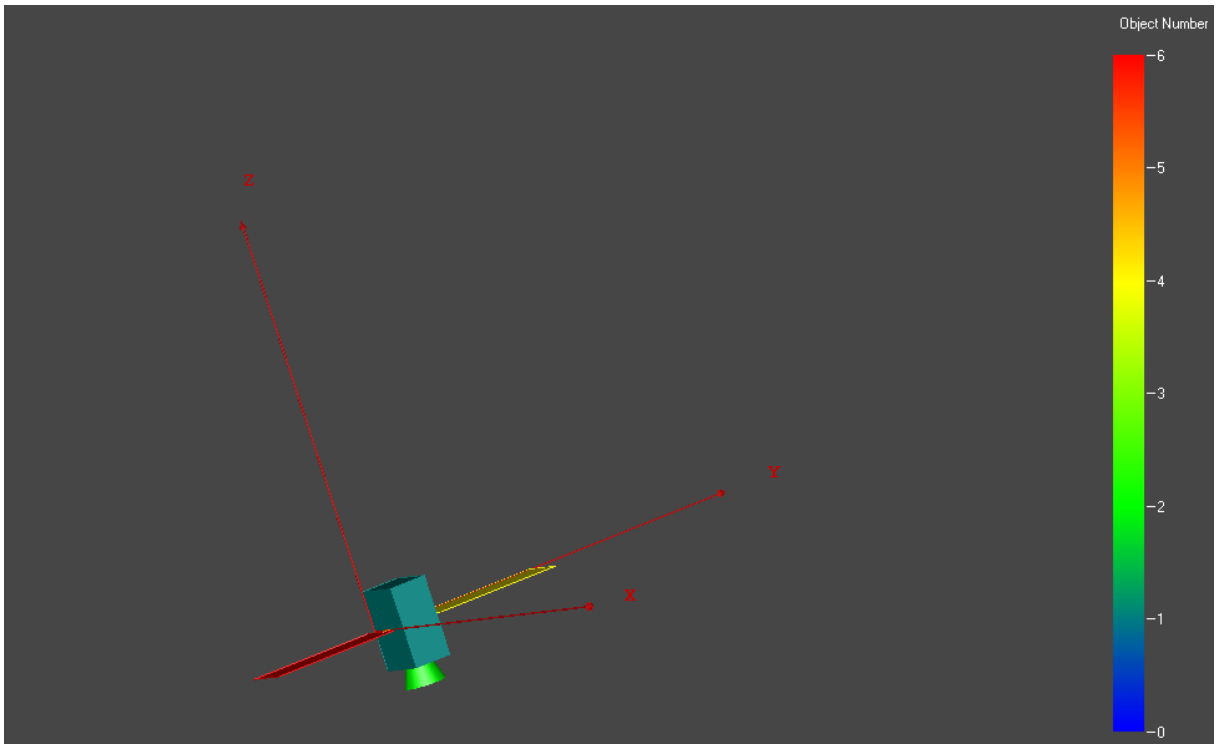

Figure 3-60: A simple satellite structure to demonstrate the STEP-SPE export

After opening the related geometry file, go to the outline and select the "System" node. Please note that the STEP-SPE export is only possible on system level, and not for single shapes.

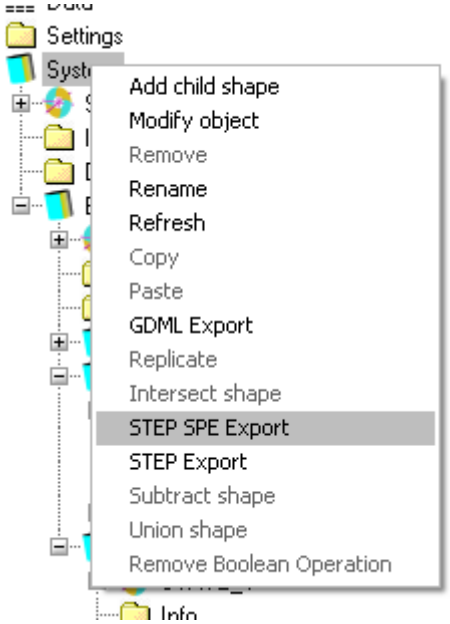

Figure 3-61: STEP-SPE export in the outline

After selecting the entry, a new window opens where you can enter the name and the location of the file which will be created to contain the geometry description in the STEP-SPE protocol. After pressing the "Save" button, the file is written to the disk. STEP-SPE is an ASCII format, so the exported file can be viewed and modified by a text editor.

In STEP-SPE, the shape definitions from STEP-TAS are used, and, if possible, ESABASE2 will map the shapes types from the geometry to the STEP-TAS types. In case this cannot be done, ESABASE2 will try to export the shape in question as a tessellated (or meshed) shape. Therefore, it will check whether the geometry was already meshed before (e. g. an analysis was performed with it), and if it finds an existing mesh, it will be exported. In case the geometry had not been meshed, it will use the OCC mesher internally to generate a mesh which can be exported. Please note that this mesh is not stored within the file. It will only be generated on-the-fly and removed after the export.

In the current implementation, all shapes from the ESABASE2 shape palette can be exported to STEP-SPE, and all unknown shape types (like imports from STEP-AP) can also be exported. It is not possible to export assemblies.

3.2.7.3 STEP-AP Export

With release 5.0 of ESABASE2, it is possible to export geometry objects to files following the STEP-AP protocol.

The export mechanism for STEP-AP follows the same principles as the GDML export described in section 3.2.7.1. We can revisit the ATV example used to show the GDML export to show how the export works in detail.

After importing the BAS file, go to the outline and select the "System" node. Please note that the STEP-AP export is only possible on system level, and not for single shapes.

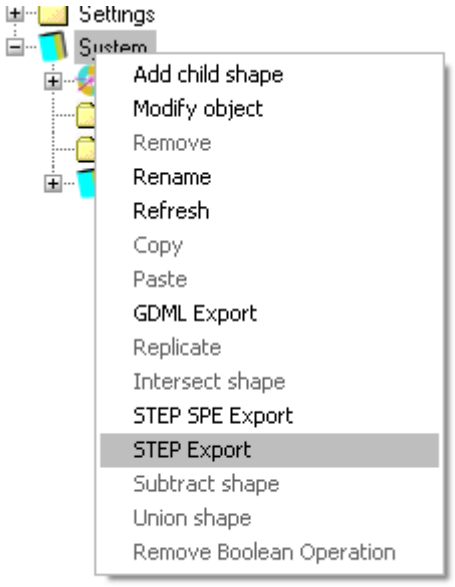

Figure 3-62: STEP export in the outline

After selecting the entry, a new window opens where you can enter the name and the location of the file which will be created to contain the geometry description in the STEP-AP protocol. After pressing the "Save" button, the file is written to the disk. STEP-AP is an ASCII format, so the exported file can be viewed and modified by a text editor.

Internally, ESABASE2 uses the OpenCASCADE Step Exporter, which maps the displayed TopoDS Shapes into the STEP-AP format. The export can be applied to all kind of geometries.

3.3 Kinematics

Kinematics was briefly discussed in tutorial style in section 3.2.3.5 and 3.2.3.6. This section discusses kinematics in more detail.

In a spacecraft geometry, the "system" is represented by the central body of the geometric model (typically the spacecraft's platform), which carries all other bodies (solar panels, antennas, rotors, etc.). The pointing of the system is directly specified on the "System" node of the geometric model.

The system is modelled in three steps:

- Establishment of the geometric objects considering the intended model hierarchy via the geometry wizards (pages "Size", "Meshing", "Position & Attitude", "Debris")
- Definition of the system's pointing via the Outline (right-click on "System", select "Modify object" from the context menu, go to the "Pointing" page and specify the required pointing directions)
- Definition of the pointing behaviour of other parts of the system via the Outline or via the 3D editor (right-click on the object to be pointed, select "Modify object" from the context menu, go to the "Kinematic" page and specify the required degrees of freedom/rotation axis/axes and their restrictions, go to the "Pointing" page and specify the intended pointing direction).

Note that all child objects of an articulated object will be oriented like their parent object, unless a different pointing is specified for parts of the geometry sub-tree or single objects which themselves have no child objects.

This section is structured into the following subsections:

- *Bodies*: Explains how shapes are connected to each other.
- *Degrees of Freedom*: Discusses the maximum allowed movement for bodies.
- *Central Body*: How kinematics applies to the central body.
- *Articulated Bodies*: How kinematics applies to non-central bodies.

3.3.1 Bodies

Each body (except the central body) is attached to another body – the so-called parent body. A body can only be attached to one parent body but a parent body can have more than one body attached to it.

All bodies are connected to the central body – either attached directly, or linked through other bodies. Therefore the orbital position of the whole system is given when the orbital position of the central body is specified. Each body is attached to its parent body through a hinge with 0, 1 or 2 rotational degrees of freedom that are used by the pointing facility to orientate the body in the manner required.

For the central body, the body reference frame is the same as the system reference frame. The orbital position of the system is the position of the origin of the central body reference frame.

The body reference frame of an articulated body is the object reference frame of the first object of that body, i.e. of the one for which the kinematic and pointing is specified.

3.3.2 Degrees of Freedom

The central body has 3 degrees of freedom. When one pointing vector (and direction) is specified for the central body, then the central body has one degree of freedom remaining, namely rotation about that pointing vector. Specifying one additional pointing vector fully determines the orientation of the central body in space.

An articulated body may have 0, 1 or 2 rotational degrees of freedom (DoF), for example:

Degrees of Freedom (DoF)	Properties/Example
0	no relative motion, i.e. rigidly connected
1	solar array, radiator
2	antenna

Table 3.2: Examples for Degree of Freedom specifications

If satisfaction of the pointing requirements for a body would require an angular position outside the allowed range, the body is rotated to the closest limit and remains there until a position within the allowed range is required.

3.3.3 The Central Body

The orientation of the central body is conditioned by a first pointing vector which is exactly pointed in the required direction. The remaining degree of freedom is used to point the central body in either of two ways:

- A second set of pointing vector and directions is specified and the pointing facility achieves the orientation of the pointing vector which has the minimum angular deviation from the specified direction.
- A body is attached to the central body using the "Inherit parent values" parameter. The remaining degree of freedom is then used such that the body's pointing vector achieves the orientation which has the minimum angular deviation from the required pointing direction.

3.3.4 Articulated Bodies

Rotational degrees of freedom are defined on the "Kinematic" page of the geometry wizards to specify how each body moves with respect to its parent (i.e. defines the direction of the relative angular motion between two linked bodies).

For pointing purposes, each articulated body may have up to two rotational degrees of freedom defined for it.

The rotation of a body may be restricted in range (minimum and maximum values) to take into account the possible mechanical constraints. If two rotational degrees of freedom are defined and there is no constraint on the rotation, the pointing is exact. However, the general case with non-orthogonal rotational degrees of freedom does not always allow exact pointing, even when no constraint has been defined. If one rotational degree of freedom is defined, the pointing minimises the angular difference between the pointing vector and the pointing direction required. This can be improved by using the degree of freedom of the parent body to get two effective degrees of freedom. In this case both the body and its parent body must have one, and only one, degree of freedom.

If, for any reason, there is no relative rotation between two bodies, zero rotational degrees of freedom may be defined.

3.4 Pointing

Following the briefly discussed Pointing in section 3.2.3.5 and 3.2.3.6, this section goes into deeper detail, in the following structure:

- *Pointing Introduction*: Why to use pointing.
- *Pointing Parameters*: How to use pointing.
- *Relationship with Mission Specification*: How pointing and orbit specification work together to determine the final pointing of a spacecraft.
- *Pointing the Bodies and Indeterminacies*: How pointing works when pointing requirements can only be partially fulfilled.
- *Pointing Output*: Describes the results of the pointing algorithm.
- *Pointing Calculation Method*: Describes the pointing algorithm.

3.4.1 Pointing Introduction

A system (for example, a spacecraft) in orbit is subject to various environmental effects, such as solar illumination, the gravitational field, the atmosphere and so on. These effects appear as forces and torques which affect the orbital position and attitude of the system (e.g. aerodynamic, radiation effects) or as material degradation (e.g. surface recession due to the atomic oxygen fluence) and depend strongly on the geometrical configuration of the system, on its orbital orientation and on the orbital position and velocity of the system.

In order to compute these effects accurately, the articulating capabilities of the bodies of the system have to be properly modelled. For example, an earth-oriented system may assume a solar array articulating capability within angular constraints in order to track the direction of the sun. With such a system, changes in the solar panel orientation with respect to the velocity vector and/or the sun significantly alter the resulting effects (e.g. torques, forces, surface degradation) on the system.

The orientation of the various bodies of an articulated system along an orbital trajectory is computed by the ESABASE2 pointing facility. The pointing facility computes the best possible pointing of each body of a configuration of an articulated system to be oriented in its required pointing direction starting with the central body.

Data for the pointing facility is specified via the "Kinematic" and "Pointing" pages of the shape wizards within the geometry editor.

The articulated system is modelled as a set of rigid bodies. It consists of a central body supporting other bodies, as shown in the following figure.

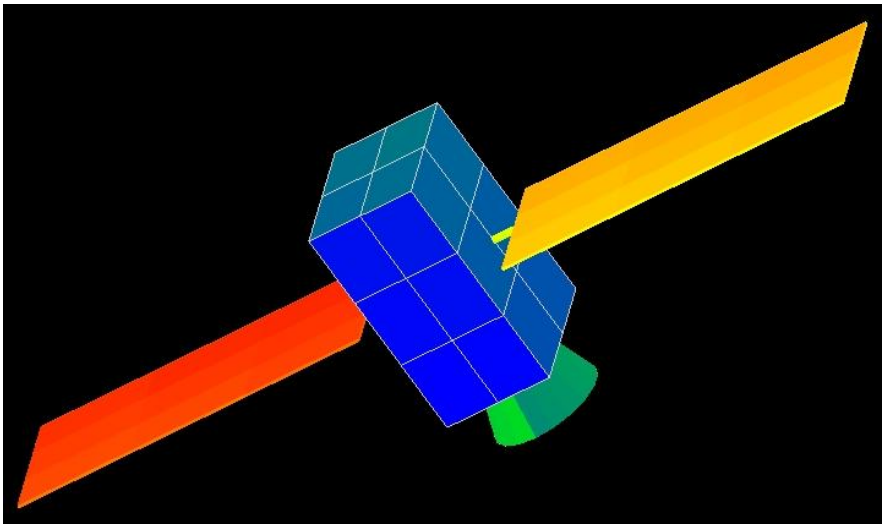

Figure 3-63: Spacecraft model

The bodies are hierarchically linked; each body imparts its own movement to the bodies that are attached to it. Each body is attached to another body through a joint with 0, 1 or 2 rotational degrees of freedom that are used by the pointing facility to orientate the body in the manner required. Constraints may be specified on the degrees of freedom, representing the rotational constraints of real joints.

The orbit of the articulated system is specified via the mission editor (see 3.2.3) by means of the orbital elements and the corresponding mission start epoch.

The pointing direction of a body in space can be described in a variety of reference frames:

- Inertial reference frames
- Sun-based reference frames
- Earth-based reference frames
- Orbit-related reference frames
- A body can also track a geo relay satellite.

The objective of the pointing facility is to orientate each body of a configuration of an articulated system (for a given orbit) in its required pointing direction, starting with the pointing of the central body.

For each degree of freedom of each body, a pointing vector (a vector always fixed relative to the body) and a pointing direction (the direction in space which the pointing vector is required to receive) are defined.

The degrees of freedom and their constraints do not always allow the required pointing direction to be achieved. On such occasions, the pointing facility gives the orientation of the pointing vector which has the minimum angular deviation – taking the constraints into account – to the required direction.

3.4.2 Pointing Parameters

This subsection describes how to apply pointing to a spacecraft geometry; it is divided into the following subsubsections:

- *General Aspects*: Things to know before considering the following pointing parameters.
- *Inherit Parent Values*: The pointing relation to the parent body.
- *VEC1 and VEC2*: The pointing vectors.
- *Direction*: The pointing direction.
- *Track*: When pointing is impossible due to eclipse, whether to follow or keep still.
- *Inverse*: When several pointing solutions are possible, which one to choose.

All parameters described below are specified via the "Kinematic" and "Pointing" pages of the shape wizards in the geometry editor. The shape wizards can be called either by rightclicking on the "System" or "Object" nodes in the geometry editor outline, or by selecting the respective object (not possible for "System") in the 3D geometry editor window and opening the context menu by rightclicking.

3.4.2.1 General Aspects

Up to fifty bodies may be pointed in ESABASE2.

The first oriented body is the central body (called "System"), which represents the overall system configuration. One body or a subtree of bodies can be defined starting at any object of the system. The chaining of these bodies is according to their hierarchical interrelation.

The Geometry Editor offers the user a shortcut ( ) to the system pointing. It can be found in the toolbar on the right side.

The pointing facility orientates each body of an articulated system in its required pointing direction, starting with the central body. If it is not possible to point the body exactly, for example, if it has only one degree of freedom, then the angle between the pointing vector and the pointing direction is minimised.

On the "Pointing" page of the geometry wizard the degrees of freedom, the pointing vectors and the pointing directions are specified. If no own kinematic and pointing is specified for a

sub-object of an articulated object, it is assumed to be rigidly connected to its parent object and will be pointed together with the parent object.

First of all, the pointing behaviour of the central body, which represents the overall system, has to be specified. All other bodies specified are connected to the central body, either directly or through a chain of other bodies.

The body reference frame of the central body is the system reference frame. The origin of this frame is positioned at the orbital points. The object reference frames of all other objects are fixed with respect to the respective object and can be translated and rotated with respect to the system reference frame as specified in the "Position & Attitude" parameters of that body.

Each body is attached to another body through a hinge with 0, 1 or 2 rotational degrees of freedom. The origin of the joint is the origin of the body reference frame.

The central body has three rotational degrees of freedom. When one pointing vector is specified for the central body, then it has one rotational degree of freedom left, namely rotation about that pointing vector. The remaining degree of freedom is used to point the central body in either of two ways:

- A second set of pointing vectors and directions is specified using "VEC2" and "pdir"; the pointing facility minimises the angle between the pointing vector and the second specified direction.
- A body is attached to the central body using the "Inherit parent values" parameter. The remaining degree of freedom is then used to minimise the angle between that body's pointing vector and its required pointing direction.

Please note that the first option is not yet available in ESABASE2.

For bodies other than the central body, the exact pointing of the pointing vector requires that two rotational degrees of freedom are present. Otherwise, the pointing will be "at best", meaning that the angle between the pointing vector and the pointing direction will be minimised.

The central body requires two pointing vectors to describe its orientation in space; the specification of its first pointing vector ("VEC1") is mandatory while the specification of its second pointing vector ("VEC2") is optional.

If the second pointing vector is not specified, the orientation of the central body must be further defined by means of the "Inherit parent values" parameter in the definition of one of the attached bodies; the remaining degree of freedom of the central body will then be used to minimise the angular difference between the pointing vector and the required pointing direction of that attached body (this attached body must itself have one degree of freedom).

The other bodies require a pointing vector and direction for each degree of freedom – the "VEC1" and "direction" parameters, respectively.

3.4.2.2 Inherit Parent Values

This parameter is accessible via the check box "Inherit parent values" on the "Pointing" page.

Parameter	Type	Range	Default	Description
parent	boolean	0 or 1	0	is a flag to indicate whether the "free" degree of freedom of the parent body will be used to minimise the angle between the pointing vector and the pointing direction of the current body.

Table 3.3: The "Inherit parent values" parameter

Parent may be specified only if:

- The body is not the central body.
- The body must have one and only one rotational degree of freedom and at least one VEC parameter.
- The parent body, if central body, has no VEC2 specified.
- The parent body, if not central body, must have one and only one rotational degree of freedom, and may not have a VEC1 parameter.
- Only one body can be attached to the parent body using the "Inherit parent values" parameter.
- No body may be attached to the current body using the "Inherit parent values" parameter.

The parent body will be orientated such that the angle between the current body's pointing vector and its' required pointing direction is minimised.

3.4.2.3 VEC1

This parameter specifies a vector in the body reference frame that will be aligned with the required pointing direction.

Parameter	Type	Range	Default	Description
vec	vector	not parallel with any axis of rotation of the degree(s) of freedom of the body	n/a	specifies a pointing vector

Table 3.4: The "VEC1" parameter

VEC1 may be specified only if:

- "VEC1" must be followed by at least one "pdir" specification.
- "VEC1" must be specified if and only if the current body:
- is the central body,
- is not the central body and has 2 rotational degrees of freedom,
- is not the central body, has 1 rotational degree of freedom, and is attached to its' parent body using "Inherit parent values".
- "VEC1" may not be specified if one other body is attached to the current body using "Inherit parent values".

3.4.2.4 VEC2

This parameter specifies a vector in the body reference frame that will be aligned with the required pointing direction, and with VEC1.

Parameter	Type	Range	Default	Description
vecb	vector	not parallel with any axis of rotation of the degree(s) of freedom of the body		specifies the additional pointing vector of the central body

Table 3.5: The "VEC2" parameter

The following restrictions apply:

- "VEC2" may only be used for the central body (the "System").
- Bodies may no longer be attached to the central body using "Inherit parent values".

"VEC2" may be specified only if:

- "VEC2" is followed by the corresponding "pdir" specification
- "VEC1" is specified.

3.4.2.5 Direction

This parameter must be specified if a "VEC1", a "VEC2" parameter is specified. It defines a pointing direction in a specified reference frame.

Parameter	Type	Range	Default	Description
dir	vector	earth, sun, velocity, georelay, fixed, central body	n/a	is a parameter specifying a pointing direction

Table 3.6: The "pdir" parameter

Six types of pointing direction specifications are made available. For a description of the reference frames available in ESABASE2 refer to 3.4.3.2.

- EARTH: The body is pointing towards the centre of the earth. This is equivalent to direction (–1,0,0) in the SUNEC reference frame.
- SUN: The body is pointing towards the sun. This is equivalent to direction (1,0,0) in the SUNEC reference frame.
- VELOCITY: The body is pointing within the orbital plane along the absolute velocity vector. This is equivalent to direction (0,1,0) in the ORBITT reference frame.
- GEORELAY: The body is pointing towards a geostationary entity, whose location is specified by its georelay longitude.
- FIXED: The body is "fixed". For the central body it implies that the orientation is fixed, in inertial space, at the last value. For all other bodies, it implies that they are rigidly connected to their parent body. If fixed is used, then track may not be specified.
- FRAME: The body's pointing direction is specified by a vector in an orbital reference frame. The vector may be specified by its coordinates or as right ascension and declination. The orbital reference frame may be inertial (INERTEQ, INERTEC) sun-based (SUNEQ, SUNEC), earth-based (EARTHE, EARTHG) or orbit related (ORBITG, ORBITT).
- CENTRAL BODY: If a complete lunar mission is analysed via an appropriate trajectory file, the central body for the first part of the mission is the Earth. When the space craft reaches a certain point of the orbit, the central body becomes the Moon. To reflect this and allow this kind of flexibility in the pointing, this frame should be used.

3.4.2.6 Track

When a pointing direction of a body is eclipsed by the earth, the body can continue tracking or stop in its latest position according to the value of the "track" flag.

Parameter	Type	Range	Default	Description

Parame-ter	Type	Range	Default	Description
track	boolean	0 or 1	1	Flag to specify the behaviour of the pointing vector when the pointing direc-tion is eclipsed by the earth.

Table 3.7: The "track" parameter

Use "track" if the pointing vector is to continue tracking the pointing direction. Disable "track" if the pointing vector is to retain the last orientation before eclipse until no longer eclipsed.

Even if exact pointing can be realised, it may be of no use when the pointing direction is eclipsed by the earth. In this case, either the pointing vector of the body continues to track the pointing direction, or the body stays in its last position.

3.4.2.7 Inverse

The "inverse" parameter is accessible via the check box on the "Pointing" page of the geometry wizards.

Parameter	Type	Range	Default	Description
inverse	boolean	0 or 1	0	Flag to indicate which body orientation to choose if two pointing solutions ex-ist. If inverse = 0 the first solution is chosen. If inverse = 1 the second solu-tion is chosen.

Table 3.8: The "inverse" parameter

If the pointing has two solutions, the model should be visualised to determine whether the solution found is the desired one. If not, then the "inverse" parameter can be used to select the second solution.

3.4.3 Relationship with Mission Specification

The pointing facility is related to the mission specified in the mission editor via:

- Orbital definition of the articulated system
- Orientation of the bodies of the articulated system in space

Since the attitude of each body is referenced to orbital specific directions such as the sun (solar panels), the earth (antennas), georelay (antennas) and the inertial direction (telescope), the following is computed:

- The system's position and the velocity which determines the orbital plane (the velocity can be considered as a special pointing direction)
- Sun ephemeris
- Earth ephemeris
- Georelay positions (if any)
- Eclipse

3.4.3.1 Orbit Definition

The definition of the orbit is described in the ESABASE User Manual /6/. The epoch is used to compute the position of the Sun, the Earth as well as the location of possible georelay satellites, and to link the pointing facility with the ESABASE debris and meteoroid models (refer to the Debris handbook).

Information whether the system is in eclipse or not is derived from the epoch parameters (time, location, velocity and eclipse flag).

3.4.3.2 Space Orientation

The orientation in space of each body of the articulated system is described by:

- A body-fixed pointing vector
- A pointing direction in a specified orbital reference frame.

A pointing direction may be specified in one of the following reference frames:

- INERTEQ: geocentric, inertial, equatorial system:
    - x within the equatorial plane, towards the vernal equinox
    - z perpendicular to equatorial plane, towards North
    - y completes the right hand system x, y, z
    - Note: INERTEQ is the so-called GAMMA-50 reference frame.

- INERTEC: geocentric, inertial, ecliptic system:
  - x within the ecliptic plane, towards the vernal equinox
  - z perpendicular to ecliptic plane, towards ecliptic North pole
  - y completes the right hand system x, y, z
- SUNEQ: geocentric, sun-fixed, equatorial system:
  - x within the equatorial plane, towards the intersection of the meridian of the true sun with the equator
  - z perpendicular to equatorial plane, towards North
  - y completes the right hand system x, y, z
- SUNEC: geocentric, sun-fixed, ecliptic system:
  - x within the ecliptic plane, towards the true sun position of date
  - z perpendicular to ecliptic plane, towards the ecliptic North pole
  - y completes the right hand system x, y, z
- EARTHE: topocentric, equatorial system:
  - x parallel to the equatorial plane, towards East
  - z perpendicular to the equatorial plane, towards North
  - y completes the right hand system x, y, z
- EARTHG: topocentric (centred at a point on earth's reference ellipsoid surface), horizontal system:
  - x within the body horizontal plane, towards East
  - y within the body horizontal plane, towards North
  - z, geodetic normal, completes the right hand system x, y, z
- ORBITG: geocentric, radius-related system (roll, pitch, yaw):
  - x within the orbital plane, directed from earth centre to the satellite
  - z perpendicular to the orbit plane in the direction of the angular momentum vector
  - y completes the right hand system x, y, z
- ORBITT: geocentric, velocity-related orbital system:
  - y within the orbital plane, along the absolute velocity vector
  - z perpendicular to the orbit plane in the direction of the angular momentum vector
  - x completes the right hand system x, y, z
- EARTHLEQ: selenocentric, Earth-fixed, tilted lunar equator system:
  - x within the orbital plane, towards the true Earth position of date
  - z perpendicular to x in the direction of the north
  - y within the equatorial plane, completes the right hand system x, y, z

To allow for more realistic modelling of the pointing behaviour of the various bodies, a time notion has been introduced for each pointing direction.

There are three ways of specifying the pointing of a body:

- as one pointing vector and one direction for all orbital positions of the body (e.g. the sun-pointing of a solar array)
- as several pointing vectors and/or several pointing directions, each pair valid for a given period of time (e.g. different targets for a space telescope) (not yet available in ESABASE2)
- as pointing directions for one pointing vector where the pointing directions are specified in an external file (e.g. a tracking antenna) (not yet available in ESABASE2)

In addition to these facilities, certain other pointing features can be modelled for the Earth orbits:

- The pointing of a body can be directed towards a georelay spacecraft by specifying a georelay longitude
- The pointing of a body can be fixed for a given period of time such that:
  - for the central body, the orientation does not change with respect to the GAMMA-50 reference frame (not yet available in ESABASE2)
  - for other bodies, the orientation of the body is fixed with respect to its parent (not yet available in ESABASE2)
- The tracking of a body can be stopped during eclipse of the pointing direction by the earth.

3.4.4 Pointing the Bodies and Indeterminacies

The pointing of the bodies in an articulated system is performed in a hierarchical manner, starting from the central body and working outwards from each parent to its children, to their children and so on. The bodies are pointed so that the orientation of their pointing vectors has the minimum angular deviation from the specified direction, but the specified pointing cannot always be achieved.

Figure 3-64 illustrates this point; if the attitude of the central body is fixed and if the hinge allows only one degree of freedom, the connected body (the solar panel) cannot be exactly orientated towards the sun. The pointing is realised by minimising the angle between the fixed body vector (the vector normal to the panel) and the pointing direction (the sun).

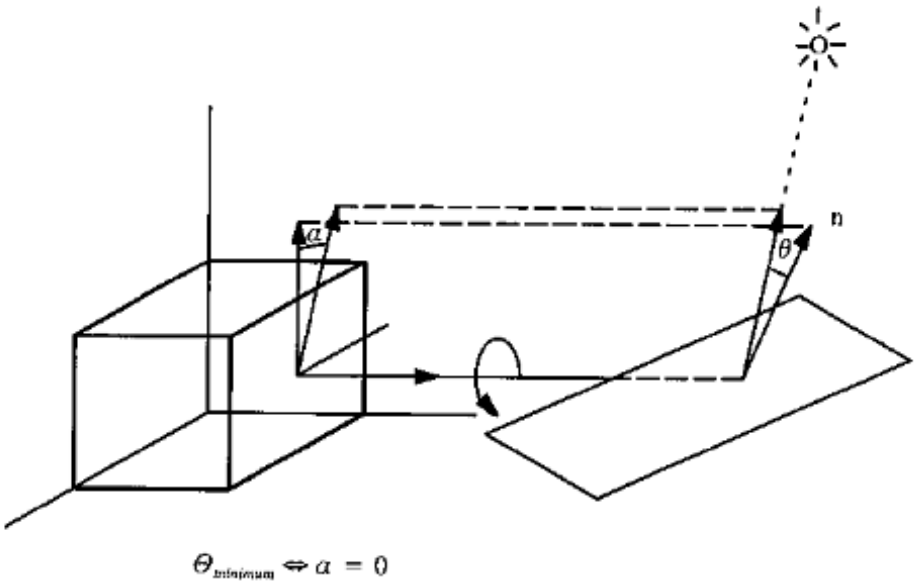

Figure 3-64: Pointing realisation

In orbit, the positions and orientations of all bodies of the articulated system must be known for a given orbital position. Hence, all degrees of freedom must be removed. This is achieved either by specifying a corresponding pointing vector and direction for the particular degree of freedom, or by letting the pointing requirements of an attached body fully determine the orientation of the current body. Any remaining degrees of freedom will be detected and an error message will be issued.

With two degrees of freedom exact orientation in a given direction is not always possible if the rotation axes (and/or one rotation axis and the pointing vector) are not perpendicular.

In two instances this causes the orientation of a body to be indeterminate; these indeterminacies are discussed in the following sections:

- *Central Body Indeterminacy*: When the central body is not fixed.

- *Body with two Degrees of Freedom Indeterminacy*: When an articulated element (a joint) is not fixed.
- *Kinematic Constraints in Pointing*: How minimum and maximum angle specified as kinematics are applied.

3.4.4.1 Central Body Indeterminacy

Figure 3-65 represents an earth-oriented platform (central body) and a connected movable body (solar array). The orientation of the platform is specified by a pointing vector p ("VEC1"). The vector p is fixed in the central body reference frame, and in this example it must point towards the earth. This condition is satisfied for any platform attitude obtained by a rotation about the p axis.

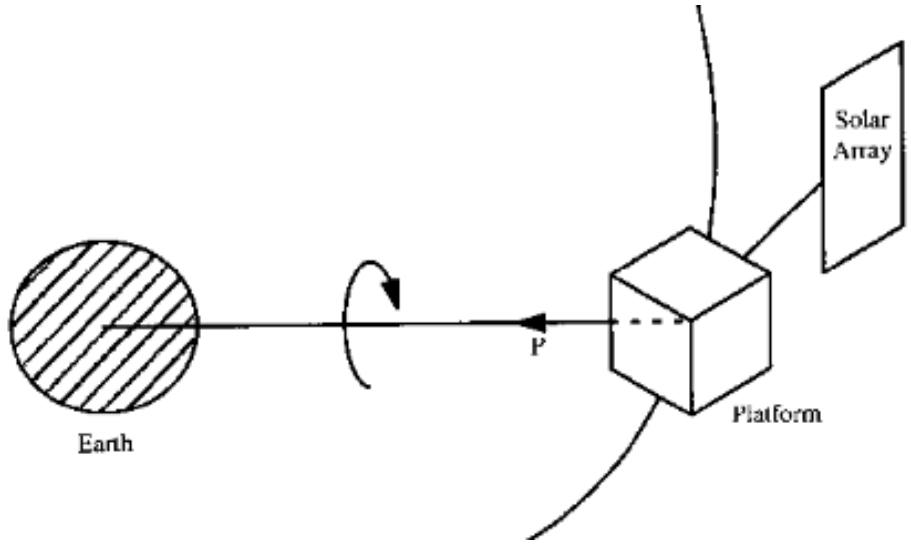

Figure 3-65: Attitude indeterminacy (central body)

The central body has 3 degrees of freedom. The exact pointing of p towards the earth removes 2 degrees of freedom. The remaining degree of freedom is the rotation about the p axis.

To fully determine the orientation of the spacecraft, the remaining degree of freedom must be removed. This may be achieved by specifying a second pointing vector, "VEC2", and a pointing direction "pdir". The central body is then rotated about p such that the pointing vector given in "VEC2" is aligned as closely as possible with the direction given in "pdir".

The second method is to let an attached body use this degree of freedom to achieve the orientation of the pointing vector which has the minimum angular deviation from the specified direction for itself, i.e. the central body will be rotated about p such that the pointing vector of the attached body achieves the minimum angular deviation from its specified pointing direction.

A second pointing vector ("VEC2") is used to suppress the indeterminacy about the first pointing vector. If this second pointing vector is not defined, the first pointing vector becomes a free axis which can be used by another body for a chained-pointing realisation (a typical case is given by the space telescope).

3.4.4.2 Body with two Degrees of Freedom Indeterminacy

This second indeterminacy case is illustrated by a simple example shown in Figure 3-66 the exact pointing is realised for two different bodies.

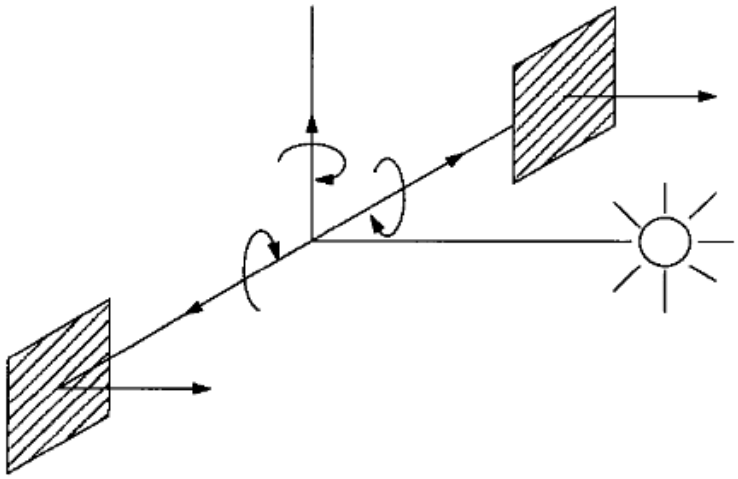

Figure 3-66: Attitude indeterminacy (2 degrees of freedom)

Physical constraints generally allow this indeterminacy to be suppressed but a special input command, provided for each body, allows one of the two solutions of a pointing with two degrees of freedom (the "Inverse" flag) to be chosen. The best way to determine whether to use this parameter is to visualise the model. The easiest way to visualise the pointing of the model is to perform an ESABASE2 run selecting the "Only pre-processing" flag in the analysis run dialog.

3.4.4.3 Kinematic Constraints in Pointing

The previous sections have implicitly assumed that the relative rotations were unrestricted. Generally, a movable body cannot rotate 360 degrees because of the mechanical constraints of the joints and/or due to the presence of other bodies.

The rotation of a body may be restricted by defining minimum and maximum angular positions with respect to its joint for each degree of freedom. If satisfaction of the pointing requirements for a body would require an angular position outside the allowed range, the body is rotated to the closest limit and remains there until a position within the allowed range is specified.

In some cases, if the constraints have not been correctly defined, the bodies may overlap one another. These cases cannot be detected by the software. Again, this can only be detected using visualisation.

3.4.5 Pointing Output

The pointing facility calculates the orientation of the pointing vector which has the minimum angular deviation from the specified direction for the bodies of the system for each of the selected orbital positions of the mission. The results of the calculations are output to the results and listing files, while the information content is the same.

For each kinematic step, the Pointing Facility outputs the following information to the report file (*.LISKIN), which is written to the "ListingFiles" folder. The report file can also be accessed through the "Listings" page of the results editor (choose the "Kinematic" listing).

A kinematic listing contains the following:

- date/time and elapsed time from the beginning of an orbital sequence
- orbit information (orbital elements, state vector and sun location) given in the inertial GAMMA-50 reference frame
- orientation of the pointing vector and the pointing direction are given in two reference frames:
    - the inertial GAMMA-50 reference frame
    - the system reference frame (i.e. the reference frame of the central body)
- relative orientation and pointing direction of the bodies

3.4.6 Pointing Calculation Method

The pointing calculation method used can be summarised as follows:

- Transform all system-related information (body reference frames and pointing vectors) to the reference frame of the central body, which represents the reference frame of the articulated system.
- Transform all orbit-related information (state vector, sun ephemerides, georelay positions and pointing directions) to the inertial equatorial GAMMA-50 reference frame.
- Convert all time-related information (epoch and pointing vectors) into modified Julian days.
- Establish the connection between the space environment (expressed in the GAMMA-50 reference frame) and the system environment (expressed in the reference frame of the central body); this is performed by pointing the central body.
- Transform the pointing directions to the reference frame of the central body.
- Transform the "pointed" articulated system and orbit-related information to the user-specified reference frame for analysis and post-processing purposes.

The calculations performed are described in the following subsubsections:

- *Initialisation part*: Activated only once, at startup.
- *Dynamic part*: Activated at each orbital position.

The calculation steps are only treated in summary form, followed by the formulas applied and any additional background information.

3.4.6.1 Initialisation Part

The "Kinematic" section of the geometry file is read and the time-related data is converted into modified Julian days. The transformation matrices between the body reference frames and that of the central body, which is identical to the system reference frame, are computed. With this information the orientation of the pointing vectors of the bodies in their initial positions are calculated in the reference frame of the central body.

The "link vectors" (i.e. the vectors between the origins of two connected bodies) are calculated and, finally, the bodies are translated so that the origins of their reference frames coincide with the origin of the central body reference frame.

3.4.6.2 Dynamic Part

This part is performed for each of the orbital points selected according to the "Pointing" parameters contained in the geometry file.

Some useful vectors are calculated in the geocentric inertial equatorial GAMMA-50 reference frame (i.e. sun ephemerides, georelay position and inertial direction) for the Earth or the bodycentric inertial frame.

The current pointing directions are determined and are transformed to the GAMMA-50 (bodycentric) reference frame. The transformation matrix between the GAMMA-50 (bodycentric) reference frame and the system reference frame is computed by pointing the central body of the geometry in the direction specified by the user.

All orbit-related information is now transformed from the GAMMA-50 (bodycentric) reference frame to the system reference frame in which the other bodies are pointed so that the orientation of the pointing vector has the minimum angular deviation from the specified direction in a hierarchical manner: the central body is pointed first and then the bodies directly attached to the central body, thereafter the bodies attached to those bodies are pointed, and so on. Special features such as "fixed" bodies and "no tracking during eclipse" are taken into account at this stage.

4 Solvers

Mission and geometry specified above can be used in a number of solvers. In order to keep the framework handbook at a reasonable size, the chapters for the solvers have been put into dedicated solver handbooks:

- The *Debris* handbook /2/ covers space debris and micro-meteoroids.
- The *Atmosphere+Ionosphere* handbook /3/ covers atmospheric and ionospheric effects; it corresponds to the Atomic Oxygen solver of the original ESABASE
- The *COMOVA* handbook /4/ covers contamination and outgassing.
- The *Sunlight* handbook /5/) covers sunlight effects.

In each handbook, input and output files as well as instructions for running an analysis are described. Geometry additions only applicable for one specific solver are also covered there, including more detailed material descriptions.

5 Troubleshooting

You can run into two kinds of trouble: The application delivers unexpected results, or the application crashes. This chapter is divided into the following sections:

- *Precautions*: How to avoid trouble.
- *Trouble Information*: Where to look for information about what is happening.
- *Application Messages*: How to interpret warnings and errors shown by the application.
- *Known Issues*: How to recognize and work around known problems.

5.1 Precautions

The following precautions can be taken to avoid trouble:

- Have as much RAM as possible (> 2 GB)
- Install the latest graphics card driver. OpenGL support must be available.
  - nvidia: http://www.nvidia.com/Download/index.aspx
  - ati: http://ati.amd.com/support/driver.html
- If you have an older graphics card or driver, avoid using two or more monitors.
- Use Nvidia or ATI graphics cards. Particularly, Intel or SiS graphics cards are not supported: Their OpenGL implementations are incomplete.

5.2 Trouble Information

There are several ways to retrieve information:

- When the application opens an error dialog, please make a screen shot (Alt+PrtScr, then paste into ImageMagick, Word or another program capable of handling images).
- The ESABASE2 log file is another valuable source of information. You can find it under the logs directory in the installed application directory. Each application start produces a new log file; the current date and time is part of the file name.

When reporting errors to us, please attach log files and a screenshot of the error message. This will shorten our error reproduction and debug times considerably.

5.3 Program Messages

Error messages of the different modules of the software are issued to different places:

- Run-time errors of ESABASE2 are displayed in a dedicated window on the screen (e.g. Figure 5-1). The application will be terminated after the occurrence of a run-time error.
- Run-time errors of the JVM or of the Eclipse platform are displayed in a dedicated window on the screen (e.g. Figure 5-2). The application will be terminated after the occurrence of such run-time errors.
- Solver error messages are displayed in a dedicated window on the screen (e.g. Figure 5-3). The behaviour of ESABASE2 depends on the type of error. In case of fatal errors, the analysis (e.g. Debris) will be terminated and the focus will be on the GUI again.

5.3.1 Messages: ESABASE2

ESABASE2 analyses are typically executed in Fortran code. With exotic data configurations, errors may be produced. Such errors will be shown as a popup window as shown in the following figure.

Visual Fortran run-time error
forrtl: severe (151): allocatable array is already allocated
Image PC Routine Line Source
DFORRT.DLL 1DAA4BD9 Unknown Unknown Unknown
DFORRT.DLL 1DAA4A39 Unknown Unknown Unknown
DFORRT.DLL 1DAA3C04 Unknown Unknown Unknown
DFORRT.DLL 1DAA348A Unknown Unknown Unknown
ESABASE.dll 1B862DDE MDGEOM 469 mdgeom.f
ESABASE.dll 1B8370A9 MAINK3AL 173 maink.f
ESABASE.dll 1B8276DB FORTRANCALLF 1109 OcafToolsWrapper.f90
TKSamples.dll 191FDF85 Unknown Unknown Unknown
OK

Figure 5-1: Troubleshooting, Messages, ESABASE2 Debris analysis

It is not possible to continue the analysis after such an error. Please check your input files for unusual configurations (e.g. a box with negative dimension in a geometry file), and make a screenshot of the error dialog when reporting the issue.

5.3.2 Messages: Eclipse and JVM

ESABASE2 relies on the Eclipse RCP framework, which in turn runs on a JVM (Java Virtual Machine). Although both have a clean exit strategy, errors caused in the OpenCascade (C++) part can lead to crashes.

The following figure shows an Eclipse/JVM runtime error.

Figure 5-2: Troubleshooting, Messages, Eclipse and JVM abnormal termination

It is not possible to recover from such an error. Examine the log file from the logs directory to detect any warnings, then restart the application. It might be helpful to proceed with fewer files being open.

5.3.3 Messages: Solver

A softer class of errors may appear within the solvers. Many invalid data definitions, for example, are detected and presented to you via error dialog.

Typical solver error messages are one of the following:

- *Pointing and kinematic*: When impossible kinematics and pointing combinations are detected.
- *Debris and other solvers*: When models are not specified for certain altitudes.

The following figure shows a solver error dialog.

Figure 5-3: Troubleshooting, Messages, Solver warning (kinematic)

In the example above, a kinematic requirement can not be fulfilled, because a degree of freedom is not available for an object.

Another example would be: "The spacecraft altitude is greater than 2000 km. Please use MASTER 2001 or 2005 for the debris analysis." The reason is that MASTER 2001 and MASTER 2005 are the only debris models which support all orbits with altitudes up to 1000 km above GEO. The other debris models are valid for orbital altitudes below 2000 km.

Generally, the error messages should be self-explaining, so you can get an idea of what went wrong. In case of doubt, send us an email.

5.4 Known Issues

The following issues are known as unresolved in ESABASE2 version 4.0.0.

- Import of complex BAS files
- Memory leak caused by Open Cascade

5.4.1 Import of Complex BAS Files

If after the import of a complex BAS-file not the entire geometric model is displayed in the 3D view or even nothing is shown, this will most probably be caused by the use of complex BAS-file constructs/ESABASE language elements. The following language elements from the original ESABASE are not available in ESABASE2:

- CONIC
- MIRROR
- NODE
- POLYNOM
- PYRAMID
- REMOVE
- REVOL
- SWEEP

We recommend replacing the above mentioned shapes with the supported shapes in the BAS file; while this is only a geometric approximation, it will ensure that the model will be correctly imported.

A second issue is the use of SHIFT and MIRROR. Although shapes are correctly imported, you should not change their values afterwards; depending on the order of BAS commands, different non-intended effects could happen. Besides, as of ESABASE2 4.0, MIRROR is no longer supported, even if not used in combination with shift.

A third issue are unsupported constructs, affecting the system and object sections. In the system section, only $DATA is supported, all other constructs are ignored:

- $ENCLOSURE
- $NODE
- $SUBSYSTEM
- $TEXT

In the object section, the following constructs are unsupported and thus ignored:

- $ANALYSIS (except RAYDENS, which is imported)
- $MASS
- within $RADIATION, INCLUDE is not supported because it requires $MASS
- $TEXT

Some sections, like $RADIATION, currently do not have a solver. The information is imported anyway, and you can access it via the Outline and Property editor (see 3.2.4), but it will not be of any use.

5.4.2 Memory Leak caused by Open CASCADE

Open CASCADE provides its own memory management which causes minor memory leaks in connection with ESABASE2 geometry handling. The consequence is cumulating memory allocation by ESABASE2, in particular when working with large geometry models.

ESABASE2 should be closed and restarted from time to time in order to avoid significant performance losses or even system crashes.

5.4.3 Meshing

ESABASE2 provides three meshers: Netgen, OCC and ESABASE (cp. section 3.2.3.2). It is observed that all three meshes are not able to mesh all available shape types correctly.

- Some meshers do not work properly for some shapes and some shapes are not supported by certain meshers.

- In the current implementation, the GDML exporter meshes the shape and exports it as a so called tessellated shape, if the shape type is unknown and the shape is meshed by the user.

- In case the shape type is unknown (i.e. in all cases where the geometry was imported via the STEP importer) and the user has not meshed it manually, the GDML exporter meshes the shape with the OCC mesher. However, in most cases the OCC mesher cannot produce correct element normal vectors. Nevertheless, the OCC mesher is used by default, because other meshers are not suitable for complex geometries.

- The import of geometries, which are generated with ESABASE2 version 3, leads normally to unmeshed geometries with OCC as default mesher for the objects.

The version 3 provided only the ESABASE mesher. A new feature can be very helpfully if the ESABASE mesher shall be applied to the objects (again). To use the feature select an object, it is not applicable to system, in the "Outline" and choose in the context menu "Apply ESABASE mesher (includes all child objects)", see next figure. This applies the ESABASE mesher with default meshing configuration to the object and its child objects.

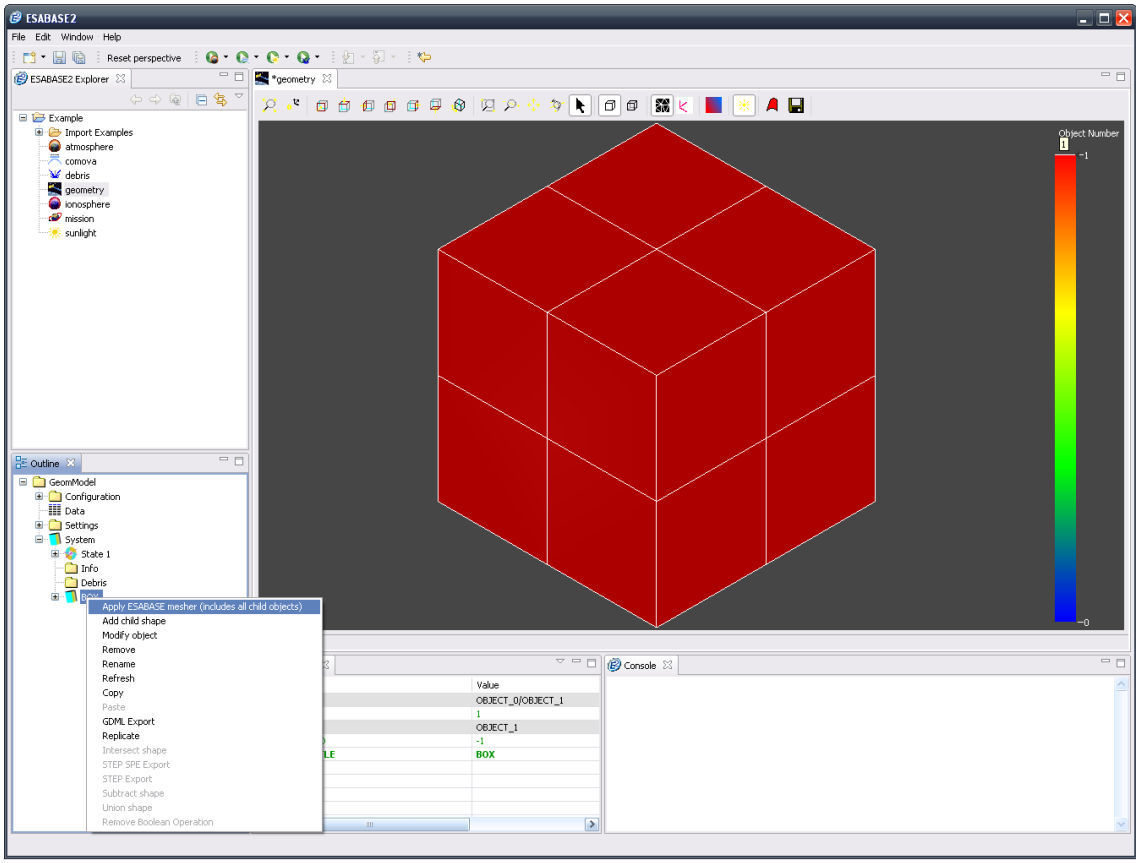

Figure 5-4: Application of the ESABASE mesher in the "Outline"

It is recommended that the user should manually mesh the geometry using the Netgen mesher before performing the export to a GDML file.

The following table shows, which shape is meshed correctly with which meshers in ESABASE2. The table uses the following terms to describe the meshing behaviour:

- *Correct*: The element normal is pointing to the correct direction.
- *Not Correct*: The element normal is not pointing to the correct direction.
- *Not Working*: The mesher does not work correctly for this shape.
- *Not Supported*: The shape is not supported by the mesher.

Shape	Netgen	OCC	ESABASE
Beam	Correct	Not Correct	Correct
Box	Correct	Correct	Correct
Cone	Correct	Not Correct	Correct
Cylinder	Correct	Not Correct	Correct
Disc	Correct	Correct	Correct
Ellipsoid	Correct	Correct (except truncated and/or segmented ellipsoid)	Correct
Parabole	Correct	Correct	Correct
Pipe	Correct	Not Currect	Correct
Plate	Correct	Correct	Correct
Prism	Correct	Not Correct	Correct
Sphere	Correct	Correct (except truncated and/or segmented sphere)	Correct
Surface	Correct	Correct	Correct
Tank	Correct	Correct	Correct
Tetrahedron	Correct	Correct	Correct
Tessellated	Correct (most cases)	Not Correct	Not Supported
Torus	Correct	Not Working	Not Supported
Trapezoid	Not Correct	Correct	Not Supported
Truss	Not Working	Not Correct	Correct

Table 5.9: Overview of the capabilities of the ESABASE2 meshers

5.4.4 ATI Graphics Card

ESABASE2 relies on the OpenCASCASDE visualisation library, which uses OpenGL to display all kinds of 3D shapes. The OpenGL implementation depends on the used graphics gard, and following OpenCASCADE's recommendations, we strongly advise to only use ESABASE2 on computers with an NVIDIA graphics card. Using it on computers with ATI graphics cards can lead to random crashes and general instabilities.

6 Annex A: Geometry Shapes

The original ESABASE framework /6/ dictates the shape palette available in ESABASE2. In addition, the Torus-, Tetrahedron- and the Trapezoid-GDML-shapes from the REAT-MS editor were integrated within the course of "Data Exchange Methods for Space Environment Tools", ESA contract number 4000101301. The following sections describe the definition and specific parameters of all available objects, structured into the subchapters:

- *Basic 3D Shapes*: Geometrical bodies which could appear in any CAD program.
- *Basic 2D Shapes*: Two-dimensional shapes with a front and back side.
- *Specific Shapes*: Flat or extruded shapes based on the "Facet" concept.
- *Complex Shapes*: Unusual bodies that aid in the construction of spacecraft.

As of ESABASE2 Release 4.0, Antenna and Ogive shapes are no longer supported. They cannot be created or imported. In addition, it is no longer possible to create or import shapes with an ellipsoid base. Both changes are due to the fact that the shape palette is now implemented using so-called OCC shapes.

6.1 Basic 3D Shapes

Basic three-dimensional geometrical shapes encompass the following:

- Box
- Cone
- Cylinder
- Ellipsoid
- Paraboloid
- Sphere

It is completely reasonable to form a first S/C approximation from these shapes; more detailed analyses would make use from more specialised shapes. The following figure gives a visual index.

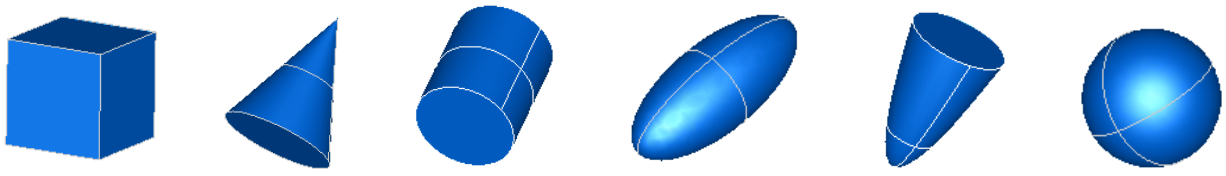

Figure 6-1: Basic 3D geometry shapes

6.1.1 Box

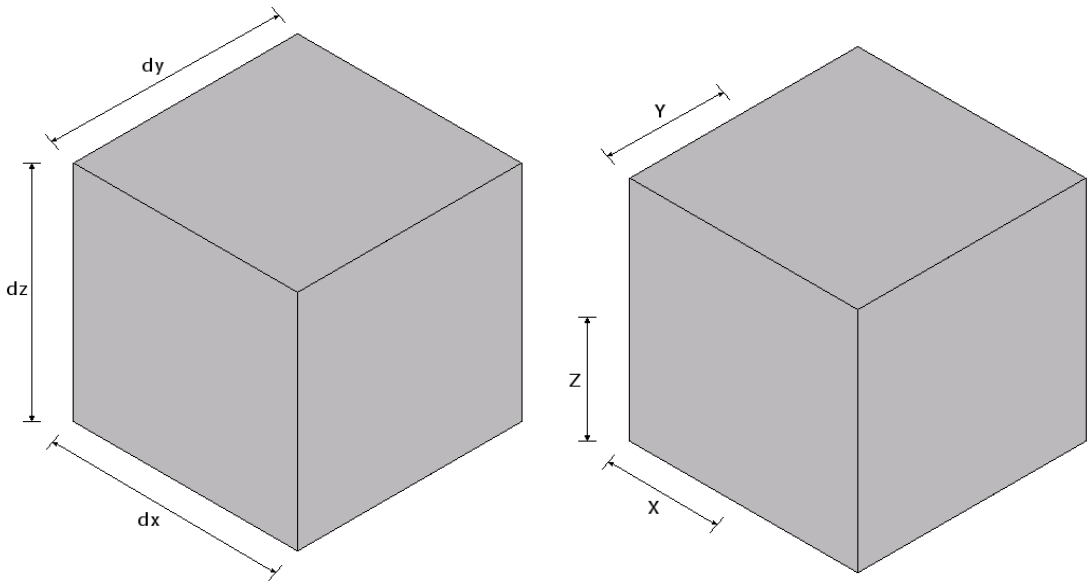

Figure 6-2: Parameters of a box object in ESABASE (left) and GDML (right) style

6.1.1.1 Purpose

To describe the shape of a rectangular box.

Parameter	Type [Unit]	Range	Default	Description
dx, dy, dz	real [mm, mm, mm]	> 0		dx, dy and dz are the lengths of the respective edges of the box along the x-, y-, and z-axes.
thick	real [mm]	0 < thick < min(dx, dy, dz)/2	solid box	Thickness of the walls of the box. If thick=0, the box is a surface shape with 6 connected faces.

Table 6.1: Definition of the BOX shape in ESABASE style

Parameter	Type [Unit]	Range	Default	Description
X, Y, Z	real [mm, mm, mm]	> 0		X, Y and Z are the half-lengths of the respective edges of the box along the

Parameter	Type [Unit]	Range	Default	Description
				x-, y-, and z-axes.

Table 6.2: Definition of the BOX shape in GDML style

6.1.1.2 Geometry and Size

This shape describes a rectangular box with its corner point (ESABASE) or its central point (GDML) at the origin of the local reference frame.

In ESABASE style the default values describe a solid box.
An enclosed cavity may be described by defining the thickness of the box:

- If the parameter "thickness" is disabled (default setting), the resulting shape is a solid with 6 faces in the following order +x, -y, -x, +y, +z, -z.
- If the parameter "thickness" is enabled and bigger than 0, the resulting shape is a volume with 12 faces: the outer faces +x, -y, -x, +y, +z, -z and the inner faces in the same order.
- If the parameter "thickness" is enabled and equals 0, the shape is a surface with 12 faces in the order: outer face +x, its inner face, outer face -y, its inner face, etc. . +x means the face in the yz plane given by the maximum x value -x means the face in the yz plane given by the minimum x value, etc. The wall are infinitely thin.

By default, the complete boundary surface of the shape is active, except if the shape has a fully enclosed cavity. In this case enclosed cavity is not active and hence not present in surface node ray-tracing analyses (e.g. ESABASE/THERMAL). If the shape is a surface shape (thickness=0) or has a fully enclosed cavity, this default can be changed via the respective drop down menu on the "Meshing" page of the shape wizard.

A box can be specified by defining the length of the x-, y- and z-axis (dx, dy, dz). The lower right corner (red dot) is the point of origin in the local reference frame (i.e. the lower right corner of the box is at (0, 0, 0)). Optionally, the user can choose to set the point of origin into the centre of the object.

By setting the cavity it is possible to get a five-sided box with one open side.

In GDML style a box can be specified by defining the length of the x-, y- and z-axis, too (X, Y, Z). But the Box extends from -X to +X, from -Y to +Y and from -Z to +Z (Similar to the ESABASE style if the point of origin was set into the centre of the object).

6.1.1.3 Meshing

The meshing directions a, b, c are defined in x-, y- and z-directions. The meshing parameter arguments are the number of elements per node area in the meshing directions a, b, c, respectively.

The numbering of nodes and elements of the shape is according to the face order and numbering directions given below, first processing the outer faces and then, if present, the inner faces:

+x face : (+y,+z)

-y face : (+x,+z)

-x face : (-y,+z)

+y face : (-x,+z)

+z face : (+y,-x)

-z face : (+y,+x)

6.1.2 Cone

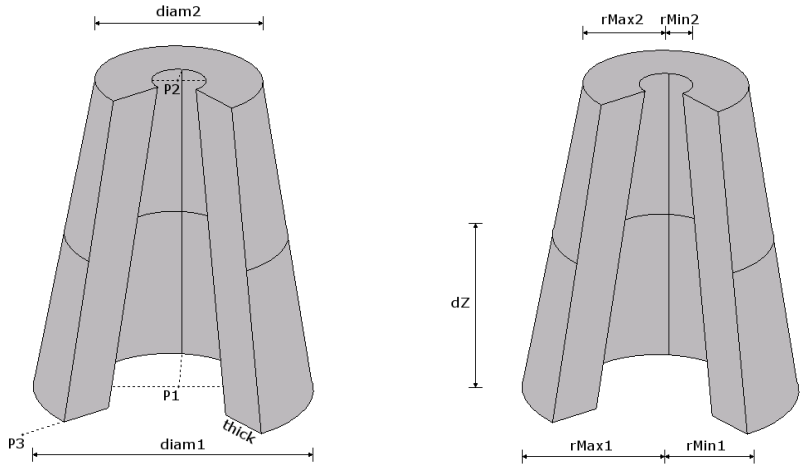

Figure 6-3: Outer shape parameters of a cone object in ESABASE (left) and GDML (right) style

6.1.2.1 Purpose

To describe the shape of either a complete or truncated circular or elliptical cone, or a section of a cone, of a given wall thickness.

Parameter	Type [Unit]	Range	Default	Description
p1	point [mm, mm, mm]			Centre of the circle at the base of the cone and the first point of the axis of revolution
p2	point [mm, mm, mm]			Centre point of the circle at the apex of the cone and specifies the second point of the axis of revolution; defined by the vector p1p2.
diam1	real [mm]	>0		External diameter of a circle at the base of the cone or external minor diameter of elliptical cone.
diam2	real [mm]	> 0	0 (complete cone)	External diameter of the circle at the apex of the cone.
thick	real [mm]	0 < thick < diam1/2	solid cone	Thickness of the cone wall perpendicular to the surface. If thick = 0 the cone is a surface.

Parameter	Type [Unit]	Range	Default	Description
p3	point [mm, mm, mm]			Optional point defining the plane p1p2p3. The intersection of this plane with the cone acts as the reference line for generating a segment, and for specifying the direction of "diame", and also for specifying the starting point of the meshing into nodes and elements. **Note:** p3 must be specified if "angle1", "angle2", "angle3" or "diame" is specified.
angle1	real [deg]	0 < angle1 < angle2 < 360	0	Start angle with reference to the plane p1p2p3 for generating a segment.
angle2	real [deg]	0 < angle1 < angle2 < 360	360	End angle with reference to the plane p1p2p3 for generating a segment.

Table 6.3: Definition of the CONE shape in ESABASE style

Parameter	Type [Unit]	Range	Description
rMin1	real [mm]	0 ≤ rMin1 < rMax1	Inside radius at the base of the cone.
rMax1	real [mm]	> 0	Outside radius at the base of the cone.
rMin2	real [mm]	0 ≤ rMin2 < rMax	Inside radius at the apex of the cone.
rMax2	real [mm]	≥ 0	Outside radius at the apex of the cone.
sPhi	real [deg]	0 ≤ sPhi ≤ 360	Starting angle of the segment in radians in the x-y-dimension.
dPhi	real [deg]	0 < dPhi ≤ 360-sPhi	Delta angle of the segment in radians in the x-y-dimension.
dZ	real [mm]	> 0	Half height of the cone in z-direction.

Table 6.4: Definition of the CONE shape in GDML style

6.1.2.2 Geometry and Size

This shape describes a circular cone or a segment of a cone. The cone may be a volume or a surface. By default, the complete boundary surface of the shape is active. For a surface shape, only one side of the boundary surface can be made active by changing this default via the respective drop down menu on the "Meshing" page of the shape wizard. Depending on the shape's parameter values, the shape's boundary surface has up to 6 faces: outer and

inner cone, base and top disc, cut faces at "angle1" and "angle2" (ESABASE style) or "sPhi" and "dPhi" (GDML style).

When defining the cone shape, the user can choose from several possible options:

- The apex of the cone can be truncated.
- The cone can be segmented vertically, i.e. like cutting a piece of cake out of it.

For a basic cone (full cone) in ESABASE style, only three parameters need to be specified. Parameter p1 defines the centre point of the base of the cone, p2 defines the centre point of the apex of the cone and "diam1" defines the diameter of the base of the cone. Because no more parameters are defined, the cone will be a full and complete cone. By default the cone is a solid cone ("thick" is empty). The thickness of the walls of the cone can be defined ("thick"). If the thickness is defined to be 0, then the walls are infinitely thin and "diam1" is the external diameter of the base of the cone.

For a basic cone in GDML style, only two parameters need to be specified. Parameter rMax1 defines the radius of the base of the cone and parameter dZ defines the half height of the cone. If the parameter rMin1 was greater than 0, the cone will be holed. The radius of the hole is rMin1.

To define a cone which is truncated parallel to the base in ESABASE style, the parameter "diam2" has to be defined. If "diam2" was 0, it would be a basic cone (with a sharp apex).

To define a truncated cone in GDML style, the parameter rMax2 has to be greater than 0. The parameter rMax2 defines the radius of the apex.

In order to segment the cone in ESABASE style, p3, "angle1" and "angle2" need to be specified. The point p3 together with p1 and p2 define the plane which serves as a starting point for the angles. Going from that point, "angle1" defines the start and "angle2" defines the end of the segment around the axis (p1, p2).

To segment the cone in GDML style, sPhi and dPhi have to be specified. Parameter sPhi defines the start angle and parameter dPhi defines the size of the segment.

6.1.2.3 Meshing

The meshing parameters are expressed along the meshing directions a, b, c (refer to the figures above).

6.1.3 Cylinder

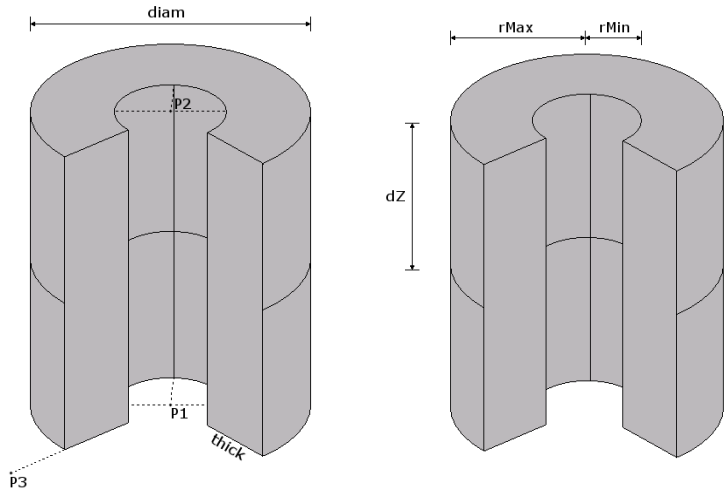

Figure 6-4: Parameters of a cylinder object in ESABASE (left) and GDML (right) style

6.1.3.1 Purpose

To describe the shape of a circular or elliptical cylinder, or a sector of a cylinder, of a given wall thickness.

Parameter	Type [Unit]	Range	Default	Description
p1	point [mm, mm, mm]			Location of the centre of the base of the cylinder and the first point of the axis of revolution.
p2	point [mm, mm, mm]			Location of the centre of the top of the cylinder and the second point of the axis of revolution; defined by the vector p1p2.
diam	real [mm]	>0		External diameter of a circular cylinder or the diameter of an elliptical cylinder perpendicular to the plane p1p2p3.
thick	real [mm]	0 < thick < diam/2	solid cylinder	Wall thickness of the cylinder. If thick = 0, the shape is a surface.
p3	point [mm, mm, mm]			Point defining the plane p1p2p3. The intersection of this plane with the cylinder acts as the

Parameter	Type [Unit]	Range	Default	Description
				reference line for generating a segment, and for specifying the direction of "diame", and also for specifying the starting point of the meshing into nodes and elements. **Note:** p3 must be specified if "angle1", "angle2", "angle3" or "diame" is specified.
angle1	real [deg]	0 < angle1 < angle2 < 360	0	Start angle with reference to the plane p1p2p3 for generating a segment.
angle2	real [deg]	0 < angle1 < angle2 < 360	360	End angle with reference to the plane p1p2p3 for generating a segment.

Table 6.5: Definition of the CYLINDER shape in ESABASE style

Parameter	Type [Unit]	Range	Description
rMin	real [mm]	≥ 0	The inner radius of the circular cylinder.
rMax	real [mm]	> 0	The outer radius of the circular cylinder.
sPhi	real [deg]	0 ≤ dPhi ≤ 360	Starting angle of the segment in radians in x-y-dimension.
dPhi	real [deg]	0 < dPhi ≤ 360-sPhi	Size angle of the segment in radians in x-y-dimension.
dZ	real [mm]	> 0	Half height of the cone in z-direction.

Table 6.6: Definition of the CYLINDER shape in GDML style

6.1.3.2 Geometry and Size

This shape describes a circular cylinder or a segment of a cylinder. The cylinder may be a volume or a surface.

By default, the complete boundary surface of the shape is active. For a surface shape, only one side of the boundary surface can be made active by changing this default via the respective drop down menu on the "Meshing" page of the shape wizard.

Depending on the shape's parameter values, the shape's boundary surface has up to 6 faces: outer and inner paraboloid, base and top disc, cut faces at "angle1" and "angle2".

For a basic cylinder (full cylinder) in ESABASE style, only three parameters need to be specified. Parameter p1 defines the centre point of the base of the cylinder, p2 defines the centre point of the top of the cylinder and "diam" defines the diameter of the base of the cylinder. Because no more parameters are defined, the cylinder will be a full and complete cylinder.

By default the cylinder is a solid cylinder ("thick" is empty). To change this, the user can define the thickness of the walls of the cylinder ("thick"). If the thickness is 0, the walls are infinitely thin. Then "diam" is the external diameter of the base of the cylinder.

For a basic cylinder in GDML style, only two parameters need to be specified. Parameter rMax defines the radius of the base of the cylinder and parameter dZ defines the half height of the cylinder. If the parameter rMin was greater than 0, the cylinder will be hollow. The inner radius is rMin.

In order to segment the cylinder in ESABASE style, the user needs to specify p3, "angle1" and "angle2". The point p3 together with p1 and p2 defines the plane which serves as starting point for the angles. Going from that point, "angle1" defines the start and "angle2" defines the end of the segment around the axis (p1, p2).

To segment the cylinder in GDML style, the parameters sPhi and dPhi have to be specified. Parameter sPhi defines the start angle and parameter dPhi defines the size of the segment.

6.1.3.3 Meshing

The meshing parameters are expressed along meshing directions a, b, c respectively. These directions are shown in Figure 6-5, along with how the elements are numbered.

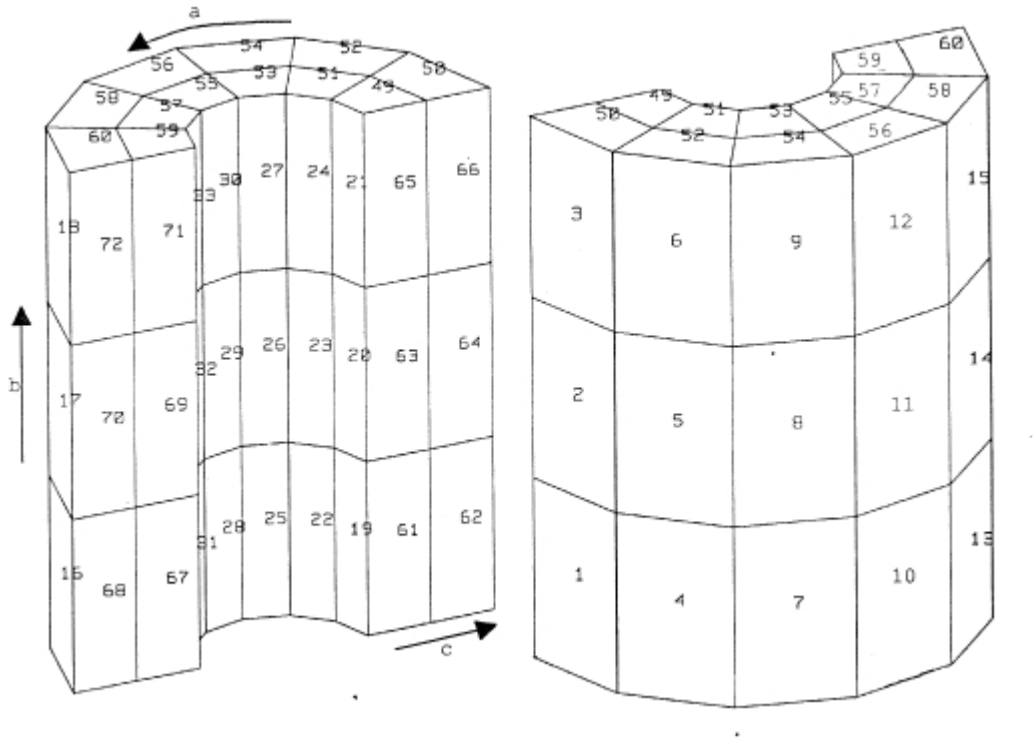

Figure 6-5: Meshing parameters of the cylinder

6.1.4 Ellipsoid

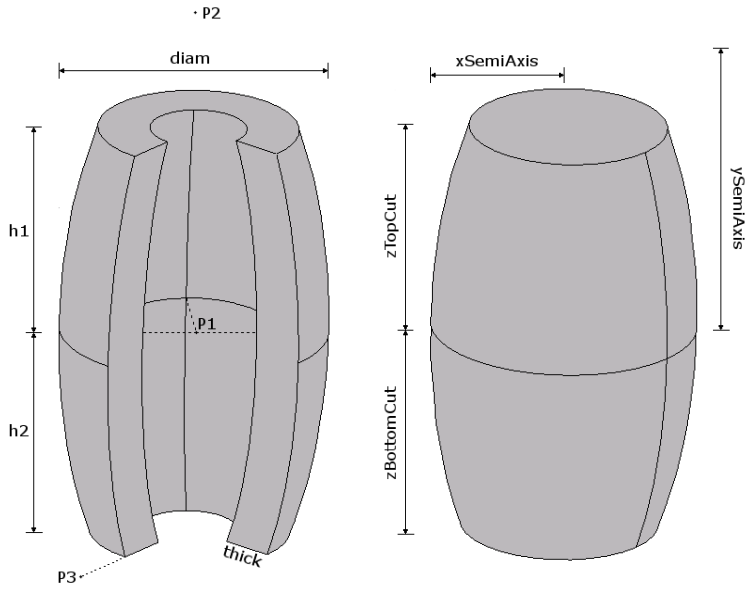

Figure 6-6: Parameters of an ellipsoid object in ESABASE (left) and GMDL (right) style

6.1.4.1 Purpose

To describe the shape of a complete circular or elliptical ellipsoid or a sector of an ellipsoid.

Parameter	Type [Unit]	Range	Default	Description
p1	point [mm, mm, mm]		0, –3000, 0	Origin of the shape and the first point of the axis of revolution.
p2	point [mm, mm, mm]		0, 3000, 0	Second point of the axis of revolution; defined by the vector p1p2.
diam	real [mm]	> 0	5000	External diameter of the circular ellipsoid, or the external minor axis of the elliptical ellipsoid, perpendicular to the plane p1p2p3 and through the point p1.
thick	real [mm]	0 < thick < min(diam/2, p1p2)	0	Wall thickness of the ellipsoid type
h1	real [mm]	$-\lvert p1p2\rvert \leq h1 < h2 + \lvert p1p2\rvert$	–6000	Location of the base of the ellipsoid with respect to point p1 along the vec-

Parameter	Type [Unit]	Range	Default	Description
				tor p1p2.
h2	real [mm]	$-\|p1p2\| \leq h1 < h2 + \|p1p2\|$	+6000	Location of the base of the ellipsoid with respect to point p1 along the vector p1p2.
p3	point [mm, mm, mm]		0, 0, 0	Point defining the plane p1p2p3. The intersection of this plane with the ellipsoid acts as the reference for generating a sector, and for specifying the direction of "diame", and also for specifying the starting point of the meshing into nodes and elements. p3 must be specified if "angle1", "angle2" or "diame" is specified.
angle1	real [deg]	0 < angle1 < angle2 < 360	0	Start angle with reference to the plane p1p2p3 for generating a segment.
angle2	real [deg]	0 < angle1 < angle2 < 360	360	End angle with reference to the plane p1p2p3 for generating a segment.

Table 6.7: Definition of the ELLIPSOID shape in ESABASE style

Parameter	Type [Unit]	Range	Description
xSemiAxis	real [mm]	> 0	Length of semi axis in X dimension.
ySemiAxis	real [mm]	> 0	Length of semi axis in Y dimension.
zSemiAxis	real [mm]	> 0	Length of semi axis in Z dimension.
zBottomCut	real [mm]	$0 \leq$ zBottomCut $\leq$ zSemiAxis	Z-coordinate for a bottom-cut. The higher the value the more gets cut of.
zTopCut	real [mm]	$0 \leq$ zTopCut $\leq$ zSemiAxis	Z-coordinate for a top-cut. The higher the value the more gets cut of.

Table 6.8: Definition of the ELLIPSOID shape GDML style

6.1.4.2 Geometry and Size

This shape describes a circular ellipsoid or a segment of an ellipsoid from "angle1" to "angle2" with a wall thickness determined by "thick".

Depending on the shape's parameter values, the shape's boundary surface has up to 6 faces: outer and inner ellipsoid, base and top disc, cut faces at "angle1" and "angle2".

By default, the complete boundary surface of the shape is active. For a surface shape, only one side of the boundary surface can be made active by changing this default via the respective drop down menu on the "Meshing" page of the shape wizard.

For a basic ellipsoid in GDML style, only two parameters need to be specified. Parameter xSemiAxis defines the radius in centre of the ellipsoid and parameter ySemiAxis defines the half length in y direction of the ellipsoid. Parameter zSemiAxis will be changed automatically with xSemiAxis.

By default the ellipsoid is a solid ellipsoid (thick is empty). To change this in ESABASE style, the thickness of the walls of the ellipsoid ("thick") can be defined. When the thickness is set to 0, then the walls are infinitely thin. If the thickness is defined, then "diam" is the external diameter of the circle around the point p1.

In order to segment the ellipsoid p3, "angle1" and "angle2" need to be specified. The points p1, p2 and p3 define the plane which serves as a starting point for the angles. Going from that point, "angle1" defines the start and "angle2" defines the end of the segment around the axis (p1, p2).

To truncate the ellipsoid h1 or h2 (ESABASE) or zBottomCut or zTopCut (GDML) must be defined. By defining h1 (zBottomCut) the bottom half of the ellipsoid can be truncated. h1 is the length from p1 to h1 along the vector(p1, p2). zBottomCut is the length from the centre to zBottomCut along the y Axis. By defining h2 (zTopCut) the top half of the ellipsoid can be truncated. h2 is the length from p1 to h2 along the vector(p1, p2). zTopCut is the legth from the centre to zTopCut along the y Axis.

6.1.4.3 Meshing

The meshing parameters are expressed along meshing directions a, b, c respectively. These directions are shown in Figure 6-7, along with how the elements are numbered.

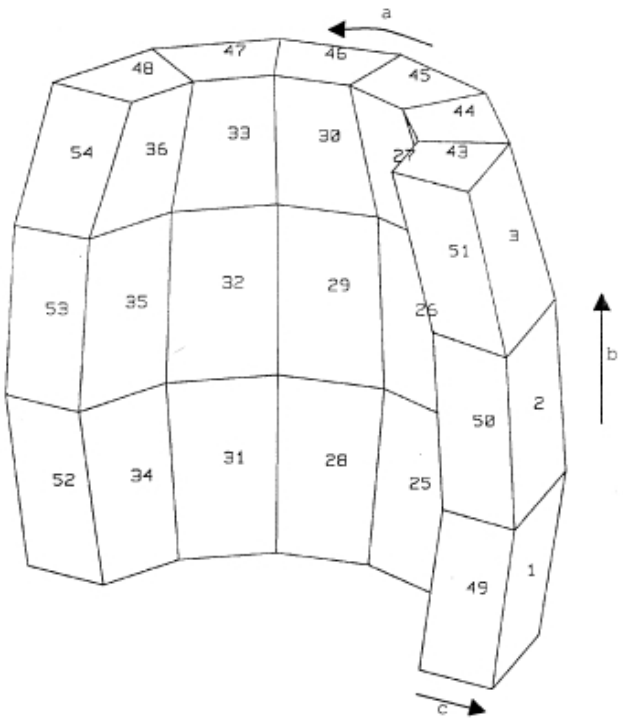

Figure 6-7: Meshing parameters of the ellipsoid

6.1.5 Paraboloid

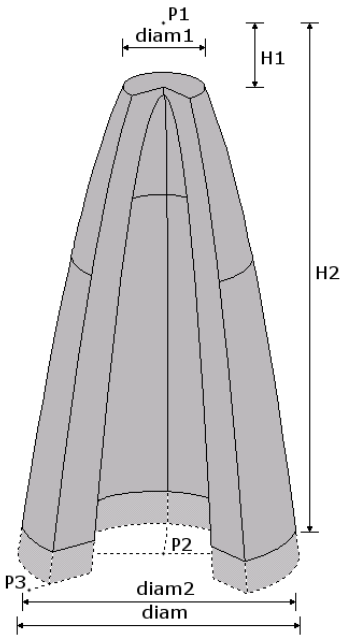

Figure 6-8: Parameters of a paraboloid object

6.1.5.1 Purpose

To describe the shape of a circular or elliptical paraboloid, or a sector of a paraboloid.

Parameter	Type [Unit]	Range	Default	Description
p1	point [mm, mm, mm]		0, –3000, 0	Origin of the shape and the first point of the axis of revolution.
p2	point [mm, mm, mm]		0, 3000, 0	Second point of the axis of revolution; defined by the vector p1p2. **Note:** p2 will be the focal point if "diam" is not specified
diam	real [mm]	> 0	4000 (point p2 is the focus)	External diameter of the circular paraboloid, or the external minor diameter of the elliptical paraboloid, perpendicular to the plane p1p2p3 and through the point p2. **Note:** If "diam" is not defined, p2 will be the focal point
thick	real [mm]	0 < thick < min(diam/2, p1p2)	solid	Wall thickness of the paraboloid. If thick=0 the paraboloid is a surface shape.

Parameter	Type [Unit]	Range	Default	Description
h1	real [mm]	> 0	0 (no truncation)	Location of the top of the paraboloid with respect to point p1 along the vector p1p2; used for generating a truncated paraboloid. **Note:** h1 may not be specified if "diam1" is specified
diam1	real [mm]	0 ≤ diam1 < diam2	0 (no truncation)	Diameter at the top of a truncated paraboloid. **Note:** "diam1" may not be specified if h1 is specified
h2	real [mm]	> 0	4000	Location of the base of the shape with respect to point p1 along the vector p1p2. **Note:** h2 may not be specified if "diam2" is specified
diam2	real [mm]	0 ≤ diam1 < diam2	equal to "diam"	Diameter at the base of the paraboloid. **Note:** "diam2" may not be specified if h2 is specified
p3	point [mm, mm, mm]		0, 0, 0	Optional point defining the plane p1p2p3. The intersection of this plane with the paraboloid acts as the reference line for generating a segment, and for specifying the direction of "diame", and also for specifying the starting point of the meshing into nodes and elements. **Note:** p3 must be specified if "angle1", "angle2" or "diame" is specified
angle1	real [deg]	0 < angle1 < angle2 < 360	0	Start angle with reference to the plane p1p2p3 for generating a segment.
angle2	real [deg]	0 < angle1 < angle2 < 360	360	End angle with reference to the plane p1p2p3 for generating a segment.

Table 6.9: Definition of the PARABOLOID shape

6.1.5.2 Geometry and Size

This shape describes a circular paraboloid or a segment of a paraboloid from "angle1" to "angle2" with a wall thickness determined by the value of "thick". The PARABOLOID shape is classed as a surface if thick=0 or a volume otherwise.

Depending on the shape's parameter values, the shape's boundary surface has up to 6 faces: outer and inner paraboloid, base and top disc, cut faces at "angle1" and "angle2".

By default, the complete boundary surface of the shape is active. For a surface shape, only one side of the boundary surface can be made active by changing this default via the respective drop down menu on the "Meshing" page of the shape wizard.

When defining the paraboloid shape, you can choose from several possible options:

- The paraboloid can be segmented vertically, i.e. like cutting a piece of cake out of it.
- The apex of the paraboloid can be truncated.

For a basic paraboloid (full paraboloid), the user needs to specify only two parameters: Parameter p1 defines the centre point of the top of the paraboloid and p2 define the centre point at base of the paraboloid. As a default "diam", which is the external diameter of the circular paraboloid base, will be 4 * | p1p2 |. Because no more parameters are defined, the paraboloid will be full and complete paraboloid. By default the paraboloid is a solid paraboloid (thick is empty). To change this, the thickness of the walls of the paraboloid (thick) needs to be specified. When the thickness is set to 0, the walls are infinitely thin. When the thickness is defined, "diam" is the external diameter of the circular base of the paraboloid.

In order to segment the paraboloid, p3, "angle1" and "angle2" have to be specified. The points p3, p1 and p2 define the plane which serves as starting point for the angles. Going from that point, "angle1" defines the start and "angle2" defines the end of the segment around the axis (p1p2).

To truncate the paraboloid, the user has to define h1, which is the length from p1 to h1, along the vector (p1p2). h2 has to be defined also, which is then the location of base of the paraboloid.

6.1.5.3 Meshing

The meshing directions a, b, c are shown in the Figure 6-9, along with how the elements are numbered.

The numbering of nodes and elements of the shape is according to the face order and numbering directions given below:

- outer/inner paraboloids: (top-base, cutl-cut2)
- base/top discs : (inner-outer, cut1 -cut2)
- cut1/cut2 faces : (inner-outer, base-top)

Note that the first value increments fastest. For a clearer picture of the node and element numbering, please refer to Figure 6-9.

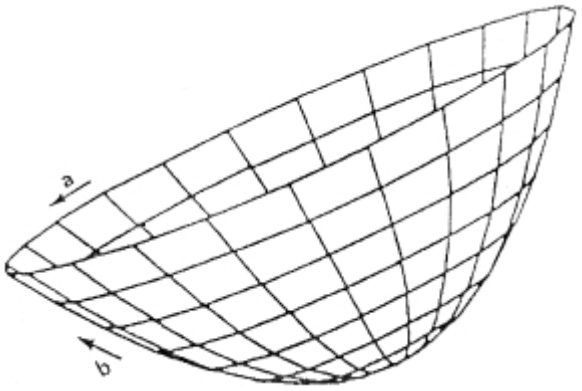

Figure 6-9: Meshing parameters and element numbering of the paraboloid

6.1.6 Sphere

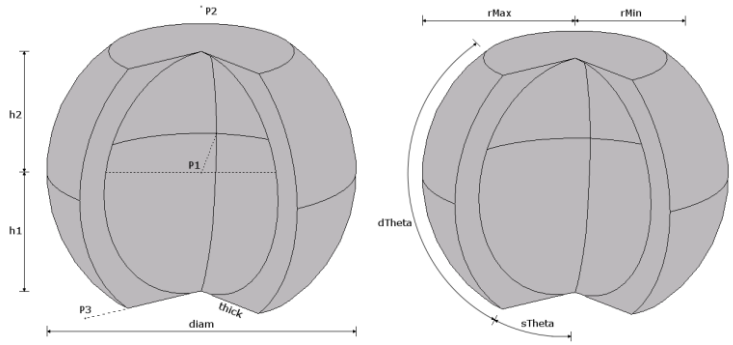

Figure 6-10: Parameters of a sphere object in ESABASE (left) and GDML (right) style

6.1.6.1 Purpose

To describe the shape of a circular or elliptical sphere or a sector of a sphere.

Parameter	Type [Unit]	Range	Default	Description
p1	point [mm, mm, mm]			Centre of the sphere and the first point of the axis of revolution.
diam	real [mm]	> 0		External diameter of the circular sphere, or the external minor diameter of the elliptical sphere, perpendicular to the plane p1p2p3 and through the point p1.
thick	real [mm]	0 < thick < diam/2	solid	Wall thickness of the sphere. If thick=0, the shape is a surface
p2	point [mm, mm, mm]			Second point of the axis of revolution; defined by the vector p1p2.
h1	real [mm]	–diam/2 ≤ h1 < h2 ≤ diam/2	–diam/2	Location of the base of the sphere with respect to point p1 along the vector p1p2 for generating a truncated sphere.
h2	real [mm]	–diam/2 ≤ h1 < h2 ≤ diam/2	diam/2	Location of the top of the sphere with respect to point p1 along the vector p1p2 for generating a truncated

Parameter	Type [Unit]	Range	Default	Description
				sphere.
p3	point [mm, mm, mm]			Optional point defining the plane p1p2p3. The intersection of this plane with the sphere acts as the reference for generating a segment, and for specifying the direction of "diame", and also for specifying the starting point of the meshing into nodes and elements.
angle1	real [deg]	0 < angle1 < angle2 < 360	0	Start angle with reference to the plane p1p2p3 for generating a segment.
angle2	real [deg]	0 < angle1 < angle2 < 360	360	End angle with reference to the plane p1p2p3 for generating a segment.

Table 6.10: Definition of the SPHERE shape in ESABASE style

Parameter	Type [Unit]	Range	Description
rMin	real [mm]	0 ≤ rMin ≤ rMax	Inner radius of the sphere – when set to 0, the sphere is solid.
rMax	real [mm]	> 0	Outer radius of the sphere.
sPhi	real [deg]	0 ≤ sPhi ≤ 360	Starting angle of the segment in radians in the x-y-plane.
dPhi	real [deg]	0 ≤ dPhi ≤ 360-sPhi	Size angle of the segment in radians in the x-y-plane.
sTheta	real [deg]	0 ≤ sTheta ≤ 180	Starting angle of the segment in radians in z dimension.
dTheta	real [deg]	0 ≤ dTheta ≤ 180-sTheta	Size angle of the segment in radians in z dimension.

Table 6.11: Definition of the SPHERE shape in GDML style

6.1.6.2 Geometry and Size

This shape describes a circular sphere or the segment of a sphere defined by "angle1" and "angle2" (ESABASE) or "sPhi" and "dPhi" with a wall thickness determined by the value of "thick". Defining parameters h1 and h2 (ESABASE) or sTheta and dTheta (GDML) allows the sphere to be truncated. The sphere shape is a surface if thick=0 or a volume otherwise.

Depending on the shape's parameter values, the shape's boundary surface has up to 6 faces: outer and inner sphere, base and top disc, cut faces at "angle1" and "angle2".

By default, the complete boundary surface of the shape is active except if the shape has a fully enclosed cavity. In this case only the positive side is active, while the enclosed cavity is not present in surface node ray-tracing analyses (e.g. EASBASE/THERMAL). If the shape is a surface shape (thick=0) or has a fully enclosed cavity, this default can be changed via the respective drop down menu on the "Meshing" page of the shape wizard.

For a basic sphere in ESABASE style (full sphere), two parameters need to be specified. Parameter p1 defines the centre point of the sphere and "diam" defines the diameter of the circular sphere. Because no more parameters are defined, the sphere will be full and complete sphere.

For a basic sphere in GDML style, only parameter rMax needs to be specified. rMax defines the outer radius of the sphere.

By default the sphere is a solid sphere ("thick" is empty). To change this in ESABASE style, the thickness of the walls of the sphere ("thick") needs to be specified. When the thickness is set to 0, the walls are infinitely thin. When the thickness is defined, "diam" is the external diameter of the circular sphere. In GDML style the difference of rMax and rMin defines the thickness of the wall of the sphere.

In order to segment the sphere in ESABASE style, p3, "angle1" and "angle2" have to be specified. The point p3 together with p1 and p2 define the plane which serves as a starting point for the angles. Going from that point, "angle1" defines the start and "angle2" defines the end of the segment around the axis (p1p2).

To segment the sphere in GDML style, sPhi and dPhi have to be specified. Parameter sPhi defines the start angle and parameter dPhi defines the size of the segment.

To truncate the sphere in ESABASE style the user has to define h1 which is the location of the base of the sphere with respect to point p1 and h2 that is the location of the top of the sphere with respect to point p1.

To truncate the sphere in GDML style sTheta and dTheta have to be specified. sTheta defines the start angle and dTheta defines size of the segment.

6.1.6.3 Meshing

The meshing directions a, b, c are shown in Figure 6-11 along with how the elements are numbered.

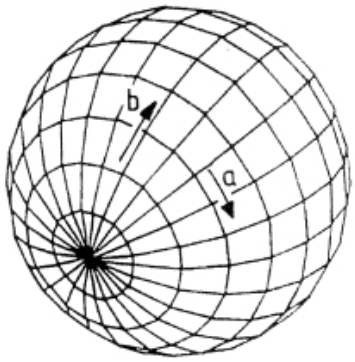

Figure 6-11: Meshing parameters of the sphere

The numbering of nodes and elements of the shape is according to the face order and numbering directions given in Figure 6-11:

- outer/inner spheres : (base-top, cut1-cut2)
- base/top discs: (inner-outer, cut 1-cut2)
- cut1/cut2 faces: (inner-outer, base-top)

Note that first value increments fastest. For a clearer picture of the node and element numbering, please refer to Figure 6-11.

6.2 Basic 2D Shapes

Apart from 3D shapes with an inside and an outside, you can also define 2D shapes with a front and back side (both of which are visible).

The following shapes are two-dimensional:

- Disk
- Plate

The figure below gives a visual index.

Figure 6-12: Basic 2D geometry shapes

6.2.1 Disc

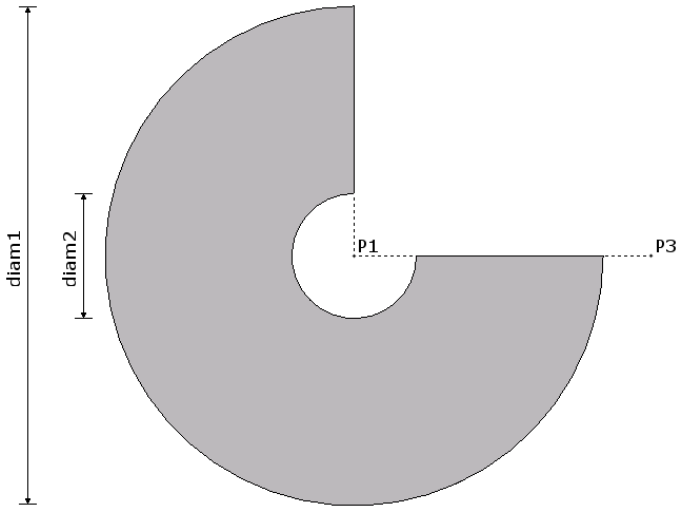

Figure 6-13: Parameters of a disc object

6.2.1.1 Purpose

To describe the shape of a circular or elliptical disc or annulus or a sector of a disc or annulus.

Parameter	Type [Unit]	Range	Default	Description
p1	point [mm, mm, mm]			Centre of the disc and the first point of the axis of revolution.
p2	point [mm, mm, mm]			Point of the axis of revolution, defined by the vector p1p2.
diam1	real [mm]	diam1 > diam2 > 0		External diameter of a circular disc, or the external minor axis of an elliptical disc, perpendicular to the vector p1p2.
diam2	real [mm]	0 < diam2 < diam1	0 (disc)	Internal diameter of a circular disc or the internal minor axis of an elliptical disc perpendicular to plane p1p2p3 for describing an annulus.

Parameter	Type [Unit]	Range	Default	Description
p3	point [mm, mm, mm]			Point defining the plane p1p2p3. The intersection of this plane with the disc acts as the reference for generating a sector, and for specifying the direction of "diame", and also for specifying the starting point of the meshing into nodes and elements.
angle1	real [deg]	0 < angle1 < angle2 < 360	0	Start angle with reference to the plane p1p2p3 for generating a sector.
angle2	real [deg]	0< angle1 < angle2 < 360	360	End angle with reference to the plane p1p2p3 for generating a segment.

Table 6.12: Definition of the DISC shape

6.2.1.2 Geometry and Size

This shape describes a circular or elliptical disc or annulus or a sector of a disc or annulus. The DISC is classed as a surface shape. By defining "diam2" an annulus can be described. A sector of a disc or annulus is described by start angle "angle1" and end angle "angle2" with reference to the plane p1p2p3.

The shape has two faces: top (on side of p2) and base. By default, both sides of the shape's boundary surface are active. This default can be changed via the respective drop down menu on the "Meshing" page of the shape wizard.

6.2.1.3 Meshing

The meshing parameters are expressed along meshing directions a and b respectively. These directions are shown in Figure 6-13, along with how the elements are numbered.

6.2.2 Plate

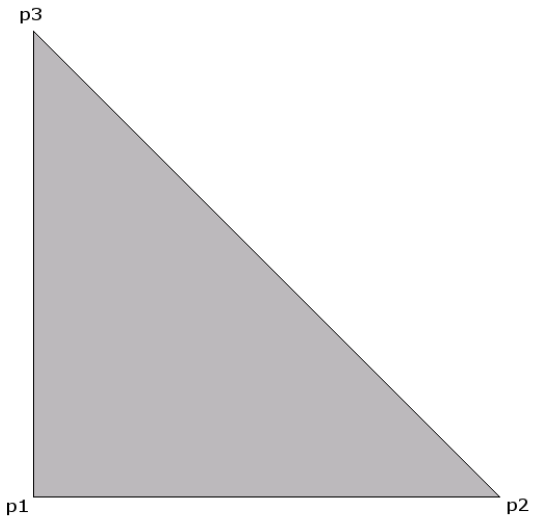

Figure 6-14: Parameters of a plate object

6.2.2.1 Purpose

To describe the shape of a planar surface between a series of 3 to 64 points.

Parameter	Type [Unit]	Range	Default	Description
p1	point [mm, mm, mm]			First point of the plate.
pn	point [mm, mm, mm]			n-th point of the plate.

Table 6.13: Definition of the PLATE shape

6.2.2.2 Geometry and Size

To describe the shape of a planar surface between a series of 3 to 64 points.

The boundary of the surface is constructed using the points in the order that they appear in the input. The attributes must be in the same plane.

The shape has two faces: top (positive side) and base.

By default, both sides of the shape's boundary surface are active. This default can be changed via the respective drop down menu on the "Meshing" page of the shape wizard.

6.2.2.3 Meshing

The meshing parameters are expressed along meshing directions a and b respectively. These directions are shown in Figure 6-15, along with how the elements are numbered.

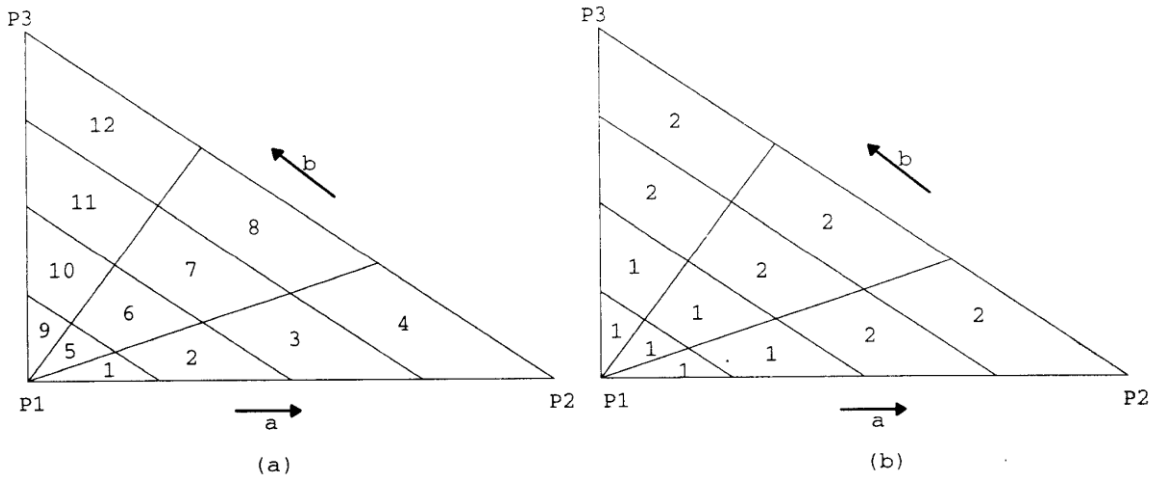

Figure 6-15: Meshing parameters of the PLATE

6.3 Specific Shapes

Specific shapes are based on the concept of a "Facet" (a free-form point list in space, which is then further extended) and encompass both 2D and 3D shapes.

The following specific shapes are based on one ore more facets (two-dimensional “dummy” shapes):

- Prism
- Surface

The figure below gives a visual index.

Figure 6-16: Specific geometry shapes

Facets will also be explained below.

6.3.1 Facet

The facet is a so-called dummy shape. It can only be used in conjunction with Surface (to build a surface-type shape) or Prism (to extrude to a 3D shape).

6.3.1.1 Purpose

To describe the shape of a planar surface between a series of 3 to 64 points.

Parameter	Type [Unit]	Range	Default	Description
p1	point [mm, mm, mm]			First point of the facet; must lie in a plane with all other points.
pn	point [mm, mm, mm]			n-th point of the facet; must lie in a plane with all other points.

Table 6.14: Definition of the FACET dummy shape

6.3.1.2 Geometry and Size

To describe the shape of a planar surface between a series of 3 to 64 points.

The surface is constructed using the points in the order that they are specified. The first and last point are implicitely joind to form the closed facet.

Facet is a dummy shape and should therefore be labelled so that it can be referred to in the enclosing shape (Surface or Prism).

6.3.1.3 Meshing

As a dummy shape, Facet has no meshing.

6.3.2 Prism

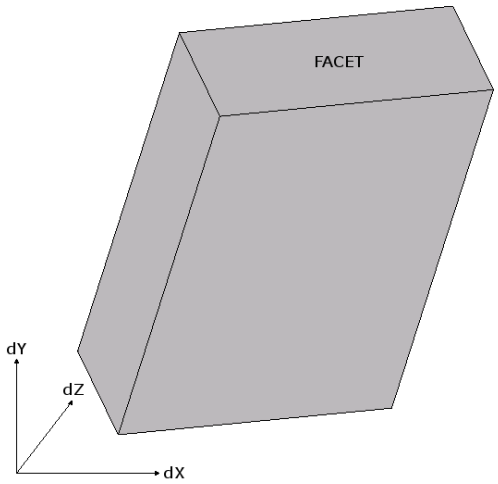

Figure 6-17: Parameters of a prism object

6.3.2.1 Purpose

To create a prismatic shape from a facet (i.e., to extrude it).

Parameter	Type [Unit]	Range	Default	Description
base	facet			A previously defined facet
dx, dy, dz	real [mm, mm, mm]	The translation vector defined by dx, dy, dz must not be in the same plane as the facet.		dx, dy and dz are translations of the facet in x, y and z directions in the object reference frame. At least one must be specified different from 0

Table 6.15: Definition of the PRISM shape

6.3.2.2 Geometry and Size

This shape describes a prism by translating a facet along a vector. The resulting shape is always a volume, as extruded from the facet.

The volumn prism has a base, top and as many quadrilateral outer faces as the base shape has sides.

By default, the complete boundary surface of the shape is active.

6.3.2.3 Meshing

The meshing parameters are expressed along meshing directions a and b respectively. These directions are shown in Figure 6-17, along with how the elements are numbered.

6.3.3 Surface

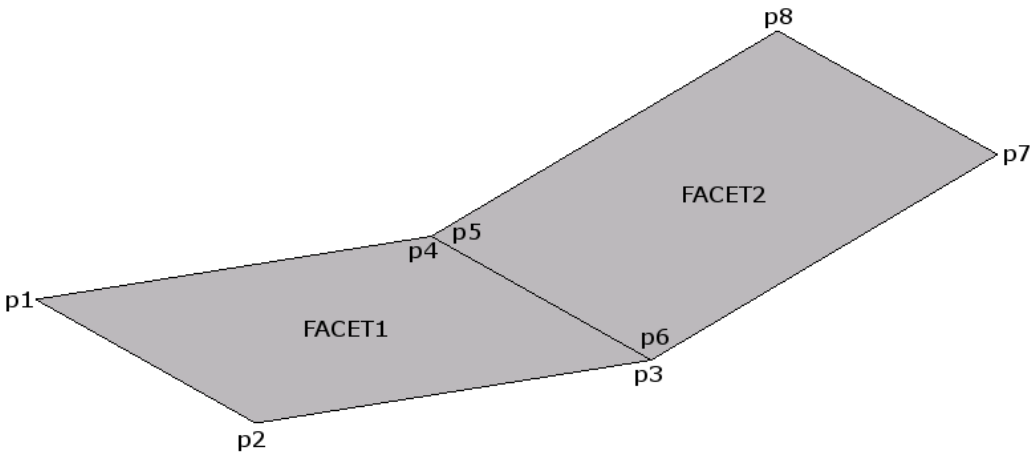

Figure 6-18: Parameters of a surface object

6.3.3.1 Purpose

To describe the shape of a polygonal surface constructed of up to 64 facets.

Parameter	Type [Unit]	Range	Default	Description
facet<n>	facet			n-th facet for the surface. Up to 64 facets are supported.

Table 6.16: Definition of the SURFACE shape

6.3.3.2 Geometry and Size

This shape describes a surface constructed of up to 64 facets.

The surface is constructed by adding the specified facets; no checking is performed on the positions of the individual shapes and the boundary lines are not removed.

The facets do not have to be connected and they do not have to be in the same plane.

By default, the complete boundary surface of the shape is active, while the positive sides are the faces with the positive normal.

6.3.3.3 Meshing

The meshing parameters are expressed along meshing directions a. These directions are shown in Figure 6-17, along with how the elements are numbered.

6.4 Complex Shapes

This section details shapes not readily found in standard CAD applications; therefore, their definition is also non-standard. They encompass:

- Beam
- Pipe
- Tank
- Truss

The following figure gives a visual index.

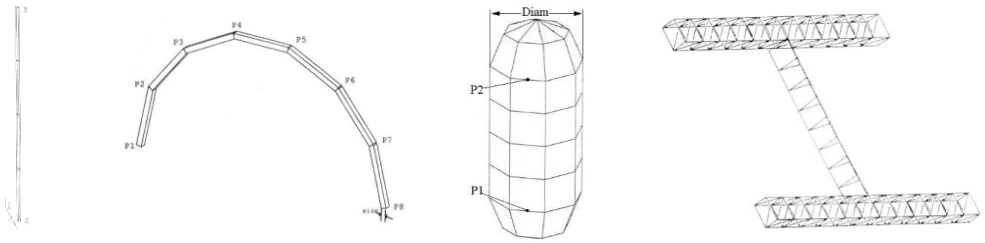

Figure 6-19: Complex geometry shapes

6.4.1 Beam

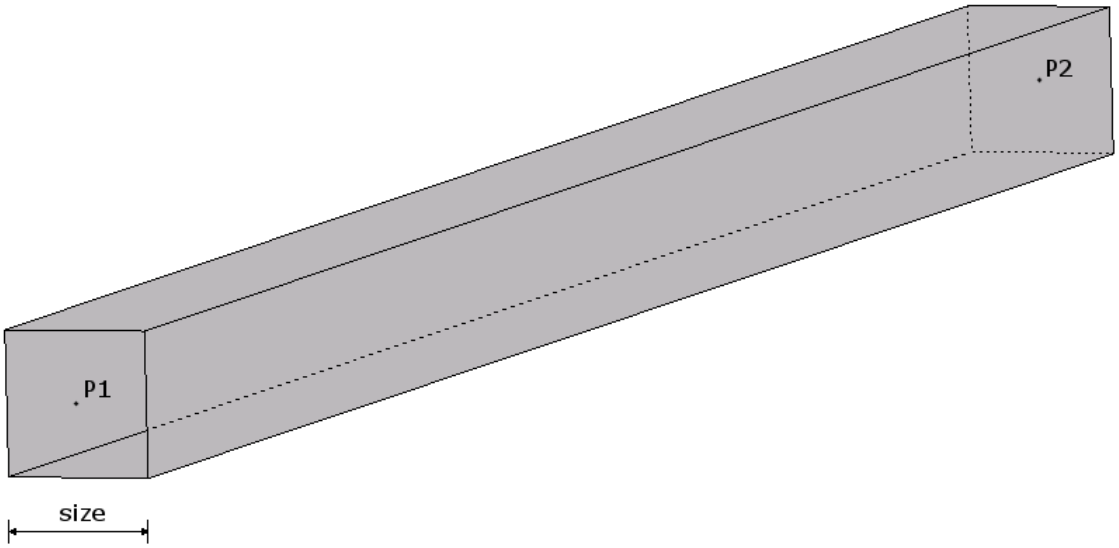

Figure 6-20: Parameters of a beam object

Figure 6-20 depicts the parameters of the beam object.

6.4.1.1 Purpose

The beam shape is described by two points p1 and p2, defined as coordinates (x, y, z). The beam will be between these two points. Additionally, the thickness of the beam ("size") can be defined. By default the thickness is 50 mm.

Parameter	Type [Unit]	Range	Default	Description
p1	point [mm, mm, mm]			End point of the beam.
p2	point [mm, mm, mm]			End point of the beam.
size	real [mm]	≥ 0	50	Length of the sides of the cross-section of the beam.

Table 6.17: Definition of BEAM shape

6.4.1.2 Geometry and Size

This shape describes a square section beam between the points p1 and p2 where p1 and p2 are the centre points of the ends of the beam. By default the beam is a line segment. By using the optional "size" parameter, the length of the sides of the square cross-section of the beam can be specified.

BEAM is classed as a line shape and hence has no surface nodes. It is not considered in surface-related analyses.

6.4.1.3 Meshing

Note that, although the shape has one or more node areas, it has no surface nodes as the shape is classed as a line shape. The meshing parameter 'a' intersects the beam into 'a' sections of equal length.

6.4.2 Pipe

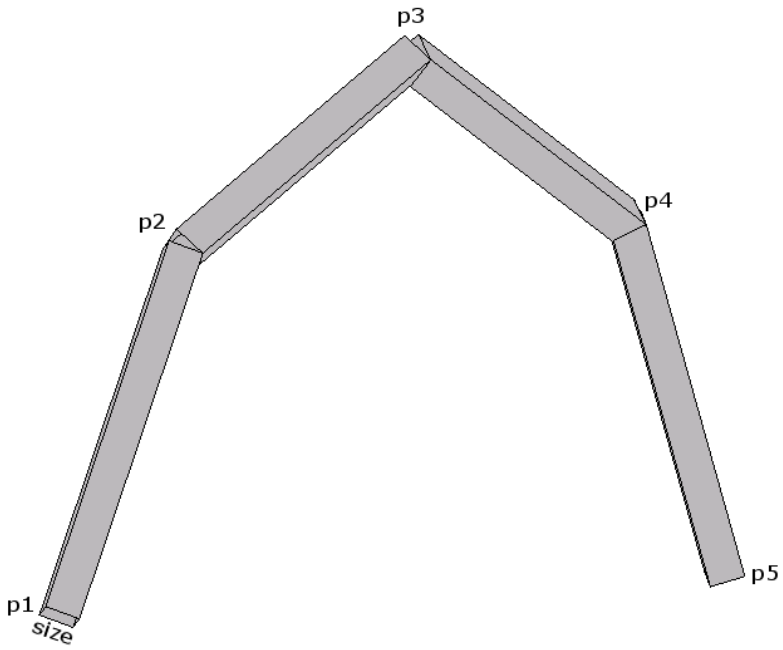

Figure 6-21: Parameters of a pipe object

Parameter	Type [Unit]	Range	Default	Description
p1, p2, p3, ..., pn	point [mm, mm, mm]			"Corner" points of the pipe. **Note:** up to 64 points may be specified.
size	real [mm]	≥ 0	0 (the pipe is a series of line segments)	Length of the sides of the cross-section of the pipe.

Table 6.18: Definition of the PIPE shape

6.4.2.1 Geometry and Size

This shape describes a square section pipe between the points p1 and p2 where p1 and p2 are the centre points of the ends of the pipe. By default the pipe is a line segment. By using the optional "size" parameter, the length of the sides of the square cross-section of the pipe can be specified.

PIPE is classed as a line shape and hence has no surface nodes. It is not considered in surface-related analyses.

In order to define a pipe, the user must specify multiple points, defined with coordinates (x, y, z). The pipe shape is a shape that will be created by connecting these points. At least two points, p1 and p2, have to be specified. Additionally the thickness of the pipe ("size") in millimetres needs to be defined. By default the thickness is 0.

6.4.2.2 Meshing

The meshing direction is from P1 to P8 as shown in the example in Figure 6-21.

Note that, although the shape has one or more node areas, it has no surface nodes as the shape is classed as a line shape.

6.4.3 Tank

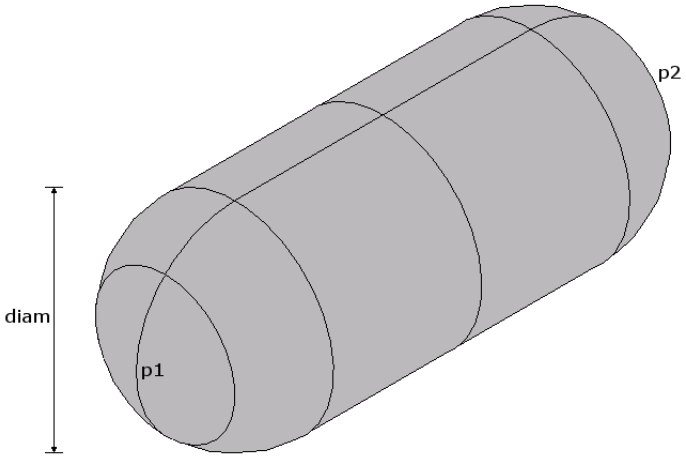

Figure 6-22: Parameters of a tank object

6.4.3.1 Purpose

To describe the shape of a "tank" consisting of a cylindrical mid-section and two hemispherical end caps.

Parameter	Type [Unit]	Range	Default	Description
p1	point [mm, mm, mm]			Location of the centre of the circle at the base of the cylindrical mid-section and the first point of the axis of revolution.
p2	point [mm, mm, mm]			Location of the centre of the circle at the top of the cylindrical mid-section and the second point of the axis of revolution, defined by the vector p1p2.
diam	real [mm]	> 0		Outer diameter of the cylindrical mid-section; this parameter also specifies the diameter of the hemispherical end caps.
thick	real [mm]	0 < thick < diam/2		Wall thickness of the tank. If thick=0, the shape is a surface.

Table 6.19: Definition of the TANK shape

6.4.3.2 Geometry and Size

This shape describes a tank, consisting of a cylindrical mid-section capped by a hemisphere at each end. The length of the mid-section of the tank is determined by the distance p1 to p2; the wall thickness of the shape is determined by the "thick" parameter. If thick=0 the shape will be a surface, otherwise a volume. Depending on the shape's parameter values, a volume tank has an outer and inner base cap, an outer and inner cylinder and an outer and inner top cap. A surface tank has an outer base cap, outer cylinder and outer top cap.

By default, the complete boundary surface of the shape is active except if the shape has a fully enclosed cavity. In this case only the positive side is active, while the enclosed cavity is not present in surface node ray-tracing analyses (e.g. EASBASE/THERMAL). If the shape is a surface shape (thick=0) or has a fully enclosed cavity, this default can be changed via the respective drop down menu on the "Meshing" page of the shape wizard.

6.4.3.3 Meshing

The meshing directions a and b are shown in Figure 6-23, along with how the elements are numbered.

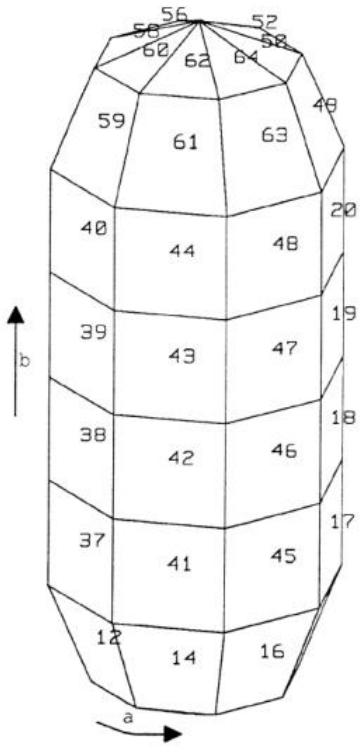

Figure 6-23: Meshing parameters of the tank

The numbering of nodes and elements of the shape is according to the face order and numbering directions given below:

- outer base cap: (base-top, anti-clockwise about p1p2)
- outer cylinder: (base-top, anti-clockwise about p1p2)
- outer top cap: (base-top, anti-clockwise about p1p2)
- inner base cap: (base-top, anti-clockwise about p1p2)
- inner cylinder: (base-top, anti-clockwise about p1p2)
- inner top cap: (base-top, anti-clockwise about p1p2)

Note that first value increments fastest. For a clearer picture of the node and element numbering, please refer to Figure 6-23.

6.4.4 Truss

Two types of truss shapes are available: the three sided truss (Truss3) and the four sided truss (Truss4).

6.4.4.1 Truss3

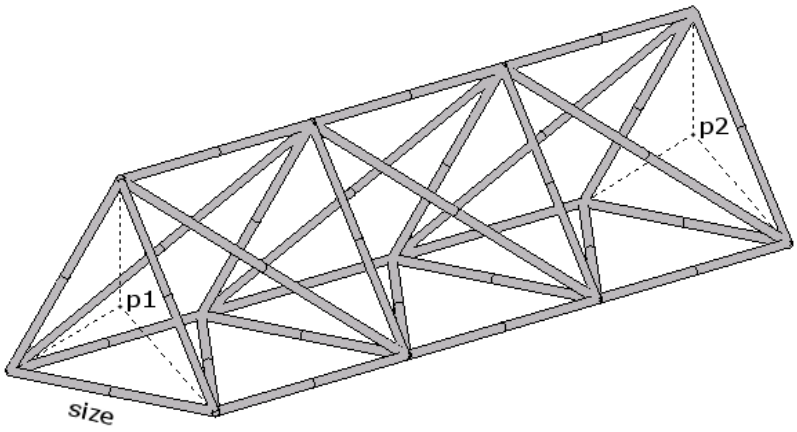

Figure 6-24: Overview of the truss object

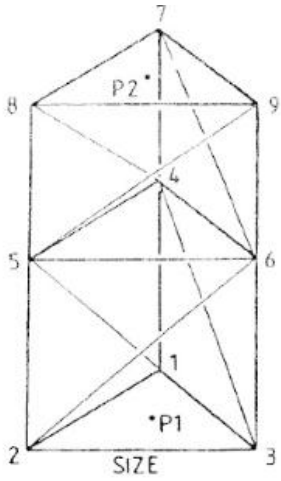

Figure 6-25: Points of the truss object

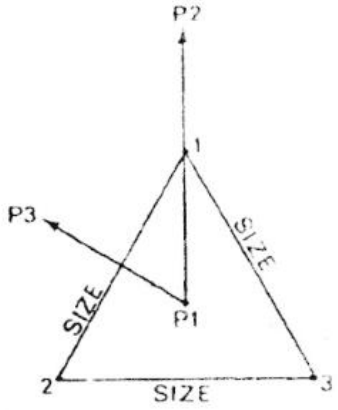

Figure 6-26: Additional parameters of the truss object

6.4.4.1.1 Purpose

To describe the shape of a triangular-section connecting truss.

Parameter	Type [Unit]	Range	Default	Description
p1	point [mm, mm, mm]			Location of the centre of one end of the truss.
p2	point [mm, mm, mm]			Location of the other end of the truss.
size	real [mm]	> 0		Length of the sides of the truss. **Note:** If size=O, the truss is treated as a line segment
nstep	int	1 ≤ nstep ≤ 50		Number of elementary sections that the shape is constructed of.
p3	point [mm, mm, mm]			Point defining the plane p1p2p3. This point defines the reference plane for orientating the truss. This plane will bisect the angle at one of the 'corners' of the triangular cross-section of the truss and one of the sides will be perpendicular to the plane.

Table 6.20: Definition of the TRUSS3 shape

6.4.4.1.2 Geometry and Size

This shape described a connecting truss whose cross-section is an equilateral triangle with the centres of the ends located at points p1 and p2. Additionally a third point p3 may be specified to define the orientation of the truss. The lengths of the sides may be specified by the "size" parameter; if size=0, the truss is a line segment.

The number of elementary sections is specified by the "nstep" parameter.

The TRUSS3 is classed as a line shape and hence has no surface nodes. It is not considered in surface related analyses.

6.4.4.1.3 Meshing

A meshing parameter cannot be applied to a TRUSS3.

6.4.4.2 Truss4

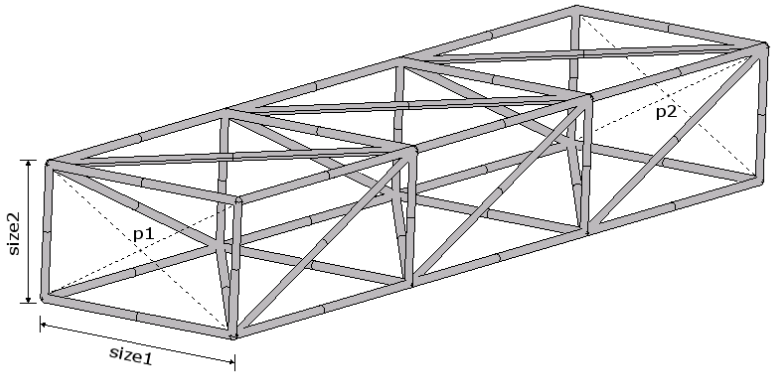

Figure 6-27: Overview of the four sided truss object

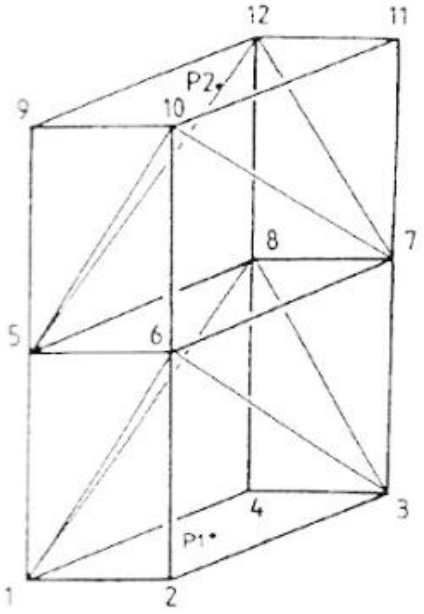

Figure 6-28: Points of the four sided truss object

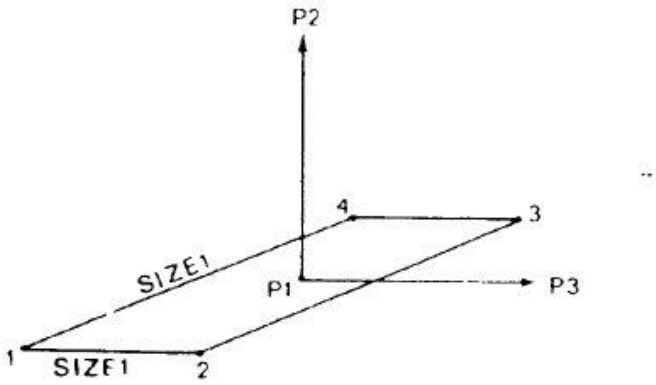

Figure 6-29: Additional parameters of the four sided truss object

6.4.4.2.1 Purpose

To describe the shape of a rectangular-section connecting truss.

Parameter	Type [Unit]	Range	Default	Description
p1	point [mm, mm, mm]			Location of the centre of one end of the truss.
p2	point [mm, mm, mm]			Location of the other end of the truss.

Parameter	Type [Unit]	Range	Default	Description
size1	real [mm]	> 0		Length of one side of the cross-section of the truss. **Note:** If size1=0, the truss is treated as a line segment
nstep	int	1 ≤ nstep ≤ 50		Number of elementary sections that the shape is constructed of.
p3 (opt)	point [mm, mm, mm]			Point defining the plane p1p2p3. This point defines the reference plane for orientating the truss. This plane will be parallel to the side of the truss whose length is indicated by "size1".
size2 (opt)	real [mm]	> 0	size2 = size1 (square cross-section)	Length of the other side of the cross-section of the truss, perpendicular to the other side and to the plane p1p2p3.

Table 6.21: Definition of the TRUSS4 shape

6.4.4.2.2 Geometry and Size

In order to define a truss4, which is a connecting truss whose cross-section is a rectangular, you must first specify two points, p1 and p2, defined as coordinates (x, y, z). These points are the centres of the ends of the truss.

Additionally the "size1", which specifies the length of the side of the truss, and "nstep" which is the number of elementary sections that the shape is constructed of, have to be defined.

Furthermore p3, which defines the orientation of the truss and "size2" which is the size of the other side of the cross-section of the truss, can be defined. If "size2" is not specified, it defaults to "size1".

6.4.4.2.3 Meshing

A meshing parameter cannot be applied to a TRUSS3 or TRUSS4 object.

7 Annex B: Additional GUI Elements

This annex describes GUI elements which are not central to the usage of ESABASE2, and therefore have not been included in either the mission/geometry or solver chapters. In particular, these are:

- *Application Menubar*: A complete listing of menubar options.
- *Geometry Editor Toolbar*: Additional options in the toolbar, e.g. for lighting or wireframe mode.

7.1 Application Menubar

The menu bar contains mostly predictable entries, as the following figure shows; only the "Window" menu will probably seem unfamiliar to you.

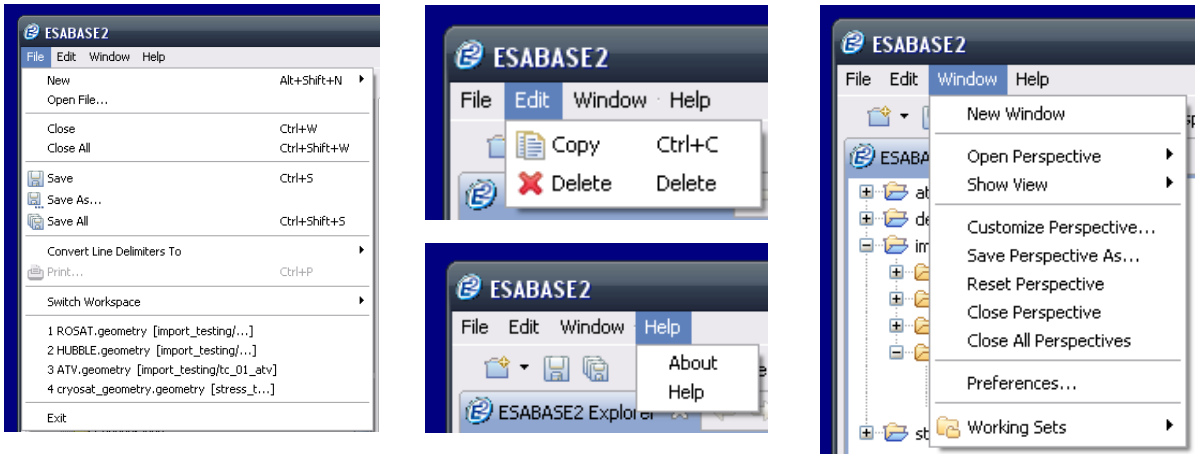

Figure 7-1: ESABASE2 menu bar

The File menu contains the usual "New", "Open", "Save", "Close" and "Exit" entries with its variants. The "Convert Line Delimiters To" allows you to convert text files from Windows to Unix and vice versa. With "Switch workspace" you can specify a non-default workspace location, though this is not recommended.

In the Edit menu, you find "Copy" and "Delete". A "Paste" operation is not included; it will appear only in context menus above the appropriate places (e.g. geometrical shapes or nodes in the Outline).

The Help menu contains an "About" box yielding among others the version of ESABASE2 you are using, and the integrated Help.

The Window menu allows you to customize your "Perspective", which is the current layout of the views you see in the application window. You can actually have more than one perspec-

tive, although you will probably not use this. With the "Show view" entry, you can control which views will be show (e.g. only explorer and editor area are possible).

In addition, the Window menu gives you access to the Preferences, in which you can modify various aspects of running ESABASE2.

Finally, the Help menu provides an About dialog and a link to the help system; the latter is depicted in the figure below.

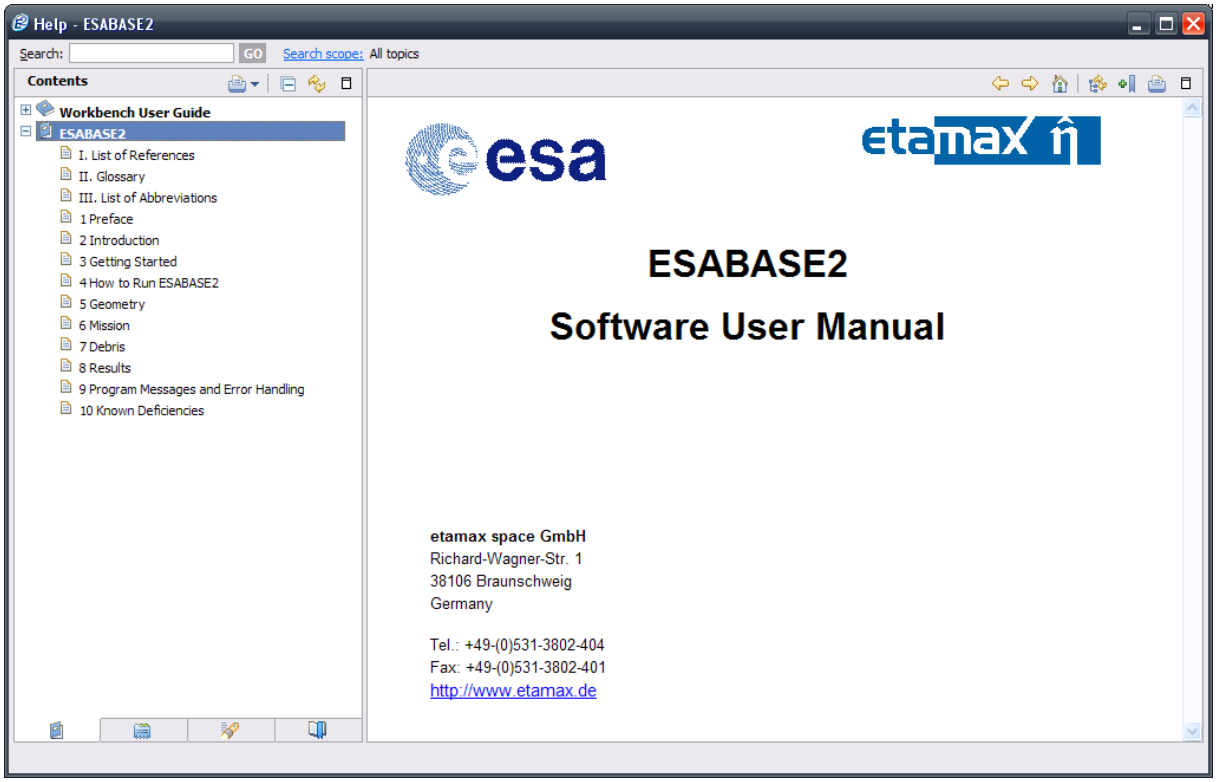

Figure 7-2: ESABASE2 help system

The help system is the same as this user manual.

7.2 Geometry Editor Toolbar

In the geometry editor, the toolbar at the top allows you to select the viewing options. The more central buttons were described in section 3.2.2.

Additional viewing mode toggle button are shown by the following figure.

Figure 7-3: Geometry editor, Toolbar (right side)

You have the choice among the following viewing options:

- Line/Wireframe mode — The objects are displayed using lines.
- Solid mode — The objects are displayed coloured with solid faces.
- Background colour — Change the background colour of the view from black to anything else. You can also change the colour of the colour scale.
- Light on/off — Enable or disable the lighting of the scene. Lighting lets models look more realistic, but will obstruct exact object colouring.